NOVEL POLYNUCLEOTIDES AND METHOD OF USE THEREOF

FIELD OF THE INVENTION

The present invention relates generally to the identification and isolation of novel nucleic acid molecules which constitute at least a portion of full-length cDNA molecules that encode human polypeptides.

BACKGROUND OF THE INVENTION

Extracellular proteins play important roles in, among other things, the formation, differentiation and maintenance of multicellular organisms. The fate of many individual cells, e.g., proliferation, migration, differentiation, or interaction with other cells, is typically governed by information received from other cells and/or the immediate environment. This information is often transmitted by secreted polypeptides (for instance, mitogenic factors, survival factors, cytotoxic factors, differentiation factors, neuropeptides, and hormones) which are, in turn, received and interpreted by diverse cell receptors or membrane-bound proteins. These secreted polypeptides or signaling molecules normally pass through the cellular secretory pathway to reach their site of action in the extracellular environment.

Secreted proteins have various industrial applications, including as pharmaceuticals, diagnostics, biosensors and bioreactors. Most protein drugs available at present, such as thrombolytic agents, interferons, interleukins, erythropoietins, colony stimulating factors, and various other cytokines, are secretory proteins. Their receptors, which are membrane proteins, also have potential as therapeutic or diagnostic agents. Efforts are being undertaken by both industry and academia to identify new, native secreted proteins. Many efforts are focused on the screening of mammalian recombinant DNA libraries to identify the coding sequences for novel secreted proteins. Examples of screening methods and techniques are described in the literature [see, for example, Klein et al., Proc. Natl. Acad. Sci., 93:7108-7113 (1996); U.S. Patent No. 5,536,637)].

Membrane-bound proteins and receptors can play important roles in, among other things, the formation, differentiation and maintenance of multicellular organisms. The fate of many

individual cells, e.g., proliferation, migration, differentiation, or interaction with other cells, is typically governed by information received from other cells and/or the immediate environment. This information is often transmitted by secreted polypeptides (for instance, mitogenic factors, survival factors, cytotoxic factors, differentiation factors, neuropeptides, and hormones) which are, in turn, received and interpreted by diverse cell receptors or membrane-bound proteins. Such membrane-bound proteins and cell receptors include, but are not limited to, cytokine receptors, receptor kinases, receptor phosphatases, receptors involved in cell-cell interactions, and cellular adhesin molecules like selectins and integrins. For instance, transduction of signals that regulate cell growth and differentiation is regulated in part by phosphorylation of various cellular proteins. Protein tyrosine kinases, enzymes that catalyze that process, can also act as growth factor receptors. Examples include fibroblast growth factor receptor and nerve growth factor receptor.

Membrane-bound proteins and receptor molecules have various industrial applications, including as pharmaceutical and diagnostic agents. Receptor immunoadhesins, for instance, can be employed as therapeutic agents to block receptor-ligand interactions. The membrane-bound proteins can also be employed for screening of potential peptide or small molecule inhibitors of the relevant receptor/ligand interaction. Efforts are being undertaken by both industry and academia to identify new, native receptor or membrane-bound proteins. Many efforts are focused on the screening of mammalian recombinant DNA libraries to identify the coding sequences for novel receptor or membrane-bound proteins.

Recently, significant progress has been made in identifying and isolating unique nucleic acid moelcules which encode all or a portion of many mammalian proteins. We herein describe the identification and characterization of novel polynucleotides which constitute at least partial cDNA molecules that encode various human polypeptides.

SUMMARY OF THE INVENTION

Novel polynucleotides have been identified and isolated which constitute at least partial cDNA molecules that encode human polypeptides.

In one embodiment, the invention provides an isolated nucleic acid molecule comprising any one of the nucleic acid sequences shown in the accompanying figures, or the complement thereof, or polynucleotide variants of those nucleic acid sequences as defined below.

In another embodiment, the invention provides an isolated nucleic acid molecule consisting essentially of any one of the nucleic acid sequences shown in the accompanying

figures, or the complement thereof, or polynucleotide variants of those nucleic acid sequences as defined below.

In another embodiment, the invention provides an isolated nucleic acid molecule consisting of any one of the nucleic acid sequences shown in the accompanying figures, or the complement thereof, or polynucleotide variants of those nucleic acid sequences as defined below.

In yet another embodiment, the invention provides an isolated nucleic acid molecule that comprises a nucleotide sequence having at least about 80% sequence identity, preferably at least about 81% sequence identity, more preferably at least about 82% sequence identity, yet more preferably at least about 83% sequence identity, yet more preferably at least about 84% sequence identity, yet more preferably at least about 85% sequence identity, yet more preferably at least about 86% sequence identity, yet more preferably at least about 87% sequence identity, yet more preferably at least about 88% sequence identity, yet more preferably at least about 89% sequence identity, yet more preferably at least about 90% sequence identity, yet more preferably at least about 91% sequence identity, yet more preferably at least about 92% sequence identity, yet more preferably at least about 93% sequence identity, yet more preferably at least about 94% sequence identity, yet more preferably at least about 95% sequence identity, yet more preferably at least about 96% sequence identity, yet more preferably at least about 97% sequence identity, yet more preferably at least about 98% sequence identity and yet more preferably at least about 99% sequence identity to (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a).

In another aspect, the isolated nucleic acid molecule consists essentially of a nucleotide sequence having at least about 80% sequence identity, preferably at least about 81% sequence identity, more preferably at least about 82% sequence identity, yet more preferably at least about 83% sequence identity, yet more preferably at least about 84% sequence identity, yet more preferably at least about 85% sequence identity, yet more preferably at least about 86% sequence identity, yet more preferably at least about 87% sequence identity, yet more preferably at least about 88% sequence identity, yet more preferably at least about 89% sequence identity, yet more preferably at least about 90% sequence identity, yet more preferably at least about 91% sequence identity, yet more preferably at least about 92% sequence identity, yet more preferably at least about 93% sequence identity, yet more preferably at least about 94% sequence identity, yet more preferably at least about 95% sequence identity, yet more preferably at least about 96% sequence

identity, yet more preferably at least about 97% sequence identity, yet more preferably at least about 98% sequence identity and yet more preferably at least about 99% sequence identity to (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a).

In yet another aspect, the isolated nucleic acid molecule consists of a nucleotide sequence having at least about 80% sequence identity, preferably at least about 81% sequence identity, more preferably at least about 82% sequence identity, yet more preferably at least about 83% sequence identity, yet more preferably at least about 84% sequence identity, yet more preferably at least about 85% sequence identity, yet more preferably at least about 86% sequence identity, yet more preferably at least about 87% sequence identity, yet more preferably at least about 88% sequence identity, yet more preferably at least about 89% sequence identity, yet more preferably at least about 90% sequence identity, yet more preferably at least about 91% sequence identity, yet more preferably at least about 92% sequence identity, yet more preferably at least about 93% sequence identity, yet more preferably at least about 94% sequence identity, yet more preferably at least about 95% sequence identity, yet more preferably at least about 96% sequence identity, yet more preferably at least about 97% sequence identity, yet more preferably at least about 98% sequence identity and yet more preferably at least about 99% sequence identity to (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a).

In another embodiment, the invention concerns an isolated nucleic acid molecule which comprises a nucleotide sequence that hybridizes to (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a). Preferably, hybridization occurs under stringent hybridization and wash conditions. Also, it is preferred that the isolated nucleic acid molecule is fully complementary to (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a).

In yet another embodiment, the present invention provides an isolated nucleic acid molecule which comprises at least about 10 consecutive nucleotides contained within (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a) which may find use as, for example, hybridizing oligonucleotide probes or for encoding polypeptide fragments that may optionally comprise a binding site for an antibody. In particular aspects, the isolated nucleic acid molecule is from about 10 to about 1000, about 10 to about 900, about 10 to about 800, about 10 to about 700, about 10 to about 600, about 10 to about 500, about 10 to about 400, about 10 to about 300, about 10 to about 200, about 10 to about 100,

about 10 to about 90, about 10 to about 80, about 10 to about 70, about 10 to about 60, about 10 to about 50, about 10 to about 40, about 10 to about 30 or about 10 to about 20 nucleotides in length, where the term "about" means the referenced nucleotide sequence length plus or minus 10% of that referenced length. In yet other aspects, the isolated nucleic acid molecule comprises at least about 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95 or 100 consecutive nucleotides contained within (a) the DNA molecule of any one of Figure 1 to 562, or (b) the complement of the DNA molecule of (a).

The present invention is also directed to a method of using an oligonucleotide probe having a nucleotide sequence derived from a nucleic acid molecule described herein for detecting the presence of and/or obtaining a full-length mammalian cDNA molecule from a mammalian cDNA library which encodes a mammalian polypeptide. Preferably, the mammal is human. The methods comprise the step of screening a mammalian cDNA library with one or more of the herein described oligonucleotides to detect the presence of a full-length cDNA and, optionally, obtaining the full-length cDNA from that library.

In another embodiment, the invention provides a vector comprising any of the isolated nucleic acid molecules described herein or their variants.

A host cell comprising such a vector is also provided. By way of example, the host cells may be CHO cells, *E. coli*, or yeast. A process for producing polypeptides is further provided and comprises culturing the host cells under conditions suitable for expression of a polypeptide and recovering the polypeptide from the cell culture.

In another embodiment, the invention provides isolated polypeptides encoded by any of the isolated nucleic acids described herein, wherein thise polypeptides are herein designated as SRT polypeptides.

In yet another embodiment, the invention provides antibodies which specifically bind to a polypeptide encoded by a nucleic acid molecule described herein. Preferably, the antibodies are monoclonal antibodies.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a nucleotide sequence (SEQ ID NO:1) designated herein as DNA8284.

Figure 2 shows a nucleotide sequence (SEQ ID NO:2) designated herein as DNA8328.

Figure 3 shows a nucleotide sequence (SEQ ID NO:3) designated herein as DNA8350.

Figure 4 shows a nucleotide sequence (SEQ ID NO:4) designated herein as DNA8369.

Figure 5 shows a nucleotide sequence (SEQ ID NO:5) designated herein as DNA8377.

Figure 6 shows a nucleotide sequence (SEQ ID NO:6) designated herein as DNA8456.

Figure 7 shows a nucleotide sequence (SEQ ID NO:7) designated herein as DNA8555.

Figure 8 shows a nucleotide sequence (SEQ ID NO:8) designated herein as DNA8576.

Figure 9 shows a nucleotide sequence (SEQ ID NO:9) designated herein as DNA9383.

Figure 10 shows a nucleotide sequence (SEQ ID NO:10) designated herein as DNA9840.

Figure 11 shows a nucleotide sequence (SEQ ID NO:11) designated herein as DNA10028.

Figure 12 shows a nucleotide sequence (SEQ ID NO:12) designated herein as DNA10072.

Figure 13 shows a nucleotide sequence (SEQ ID NO:13) designated herein as DNA10242.

Figure 14 shows a nucleotide sequence (SEQ ID NO:14) designated herein as DNA10281.

Figure 15 shows a nucleotide sequence (SEQ ID NO:15) designated herein as DNA12628.

Figure 16 shows a nucleotide sequence (SEQ ID NO:16) designated herein as DNA12646.

Figure 17 shows a nucleotide sequence (SEQ ID NO:17) designated herein as DNA12655.

Figure 18 shows a nucleotide sequence (SEQ ID NO:18) designated herein as DNA12660.

Figure 19 shows a nucleotide sequence (SEQ ID NO:19) designated herein as DNA12668.

Figure 20 shows a nucleotide sequence (SEQ ID NO:20) designated herein as DNA12726.

Figure 21 shows a nucleotide sequence (SEQ ID NO:21) designated herein as DNA12728.

Figure 22 shows a nucleotide sequence (SEQ ID NO:22) designated herein as DNA12729.

Figure 23 shows a nucleotide sequence (SEQ ID NO:23) designated herein as DNA12732.

Figure 24 shows a nucleotide sequence (SEQ ID NO:24) designated herein as DNA12733.

Figure 25 shows a nucleotide sequence (SEQ ID NO:25) designated herein as DNA12741.

Figure 26 shows a nucleotide sequence (SEQ ID NO:26) designated herein as DNA12742.

Figure 27 shows a nucleotide sequence (SEQ ID NO:27) designated herein as DNA12747.

Figure 28 shows a nucleotide sequence (SEQ ID NO:28) designated herein as DNA12752.

Figure 29 shows a nucleotide sequence (SEQ ID NO:29) designated herein as DNA12797.

Figure 30 shows a nucleotide sequence (SEQ ID NO:30) designated herein as DNA12801.

Figure 31 shows a nucleotide sequence (SEQ ID NO:31) designated herein as DNA12802.

Figure 32 shows a nucleotide sequence (SEQ ID NO:32) designated herein as DNA12817.

Figure 33 shows a nucleotide sequence (SEQ ID NO:33) designated herein as DNA12819.

Figure 34 shows a nucleotide sequence (SEQ ID NO:34) designated herein as DNA12829.

Figure 35 shows a nucleotide sequence (SEQ ID NO:35) designated herein as DNA12830.

Figure 36 shows a nucleotide sequence (SEQ ID NO:36) designated herein as DNA12834.

Figure 37 shows a nucleotide sequence (SEQ ID NO:37) designated herein as DNA12837.

Figure 38 shows a nucleotide sequence (SEQ ID NO:38) designated herein as DNA12840.

Figure 39 shows a nucleotide sequence (SEQ ID NO:39) designated herein as DNA12841.

Figure 40 shows a nucleotide sequence (SEQ ID NO:40) designated herein as DNA12844.

Figure 41 shows a nucleotide sequence (SEQ ID NO:41) designated herein as DNA12846.

Figure 42 shows a nucleotide sequence (SEQ ID NO:42) designated herein as DNA12850.

Figure 43 shows a nucleotide sequence (SEQ ID NO:43) designated herein as DNA12865.

Figure 44 shows a nucleotide sequence (SEQ ID NO:44) designated herein as DNA12867.

Figure 45 shows a nucleotide sequence (SEQ ID NO:45) designated herein as DNA12884.

Figure 46 shows a nucleotide sequence (SEQ ID NO:46) designated herein as DNA12889.

Figure 47 shows a nucleotide sequence (SEQ ID NO:47) designated herein as DNA12891.

Figure 48 shows a nucleotide sequence (SEQ ID NO:48) designated herein as DNA12900.

Figure 49 shows a nucleotide sequence (SEQ ID NO:49) designated herein as DNA12922.

Figure 50 shows a nucleotide sequence (SEQ ID NO:50) designated herein as DNA12946.

Figure 51 shows a nucleotide sequence (SEQ ID NO:51) designated herein as DNA12967.

Figure 52 shows a nucleotide sequence (SEQ ID NO:52) designated herein as DNA12974.

Figure 53 shows a nucleotide sequence (SEQ ID NO:53) designated herein as DNA12982.

Figure 54 shows a nucleotide sequence (SEQ ID NO:54) designated herein as DNA12983.

Figure 55 shows a nucleotide sequence (SEQ ID NO:55) designated herein as DNA12991.

Figure 56 shows a nucleotide sequence (SEQ ID NO:56) designated herein as DNA12998.

Figure 57 shows a nucleotide sequence (SEQ ID NO:57) designated herein as DNA12999.

Figure 58 shows a nucleotide sequence (SEQ ID NO:58) designated herein as DNA13101.

Figure 59 shows a nucleotide sequence (SEQ ID NO:59) designated herein as DNA13104.

Figure 60 shows a nucleotide sequence (SEQ ID NO:60) designated herein as DNA13110.

Figure 61 shows a nucleotide sequence (SEQ ID NO:61) designated herein as DNA13114.

Figure 62 shows a nucleotide sequence (SEQ ID NO:62) designated herein as DNA13115.

Figure 63 shows a nucleotide sequence (SEQ ID NO:63) designated herein as DNA13116.

Figure 64 shows a nucleotide sequence (SEQ ID NO:64) designated herein as DNA13118.

Figure 65 shows a nucleotide sequence (SEQ ID NO:65) designated herein as DNA13124.

Figure 66 shows a nucleotide sequence (SEQ ID NO:66) designated herein as DNA13132.

Figure 67 shows a nucleotide sequence (SEQ ID NO:67) designated herein as DNA13133.

Figure 68 shows a nucleotide sequence (SEQ ID NO:68) designated herein as DNA13146.

Figure 69 shows a nucleotide sequence (SEQ ID NO:69) designated herein as DNA13152.

Figure 70 shows a nucleotide sequence (SEQ ID NO:70) designated herein as DNA13156.

Figure 71 shows a nucleotide sequence (SEQ ID NO:71) designated herein as DNA13163.

Figure 72 shows a nucleotide sequence (SEQ ID NO:72) designated herein as DNA13185.

Figure 73 shows a nucleotide sequence (SEQ ID NO:73) designated herein as DNA13992.

Figure 74 shows a nucleotide sequence (SEQ ID NO:74) designated herein as DNA14523.

Figure 75 shows a nucleotide sequence (SEQ ID NO:75) designated herein as DNA14656.

Figure 76 shows a nucleotide sequence (SEQ ID NO:76) designated herein as DNA14938.

Figure 77 shows a nucleotide sequence (SEQ ID NO:77) designated herein as DNA15172.

Figure 78 shows a nucleotide sequence (SEQ ID NO:78) designated herein as DNA15618.

Figure 79 shows a nucleotide sequence (SEQ ID NO:79) designated herein as DNA16546.

Figure 80 shows a nucleotide sequence (SEQ ID NO:80) designated herein as DNA16669.

Figure 81 shows a nucleotide sequence (SEQ ID NO:81) designated herein as DNA17244.

Figure 82 shows a nucleotide sequence (SEQ ID NO:82) designated herein as DNA18382.

Figure 83 shows a nucleotide sequence (SEQ ID NO:83) designated herein as DNA18444.

Figure 84 shows a nucleotide sequence (SEQ ID NO:84) designated herein as DNA18649.

Figure 85 shows a nucleotide sequence (SEQ ID NO:85) designated herein as DNA19597.

Figure 86 shows a nucleotide sequence (SEQ ID NO:86) designated herein as DNA19601.

Figure 87 shows a nucleotide sequence (SEQ ID NO:87) designated herein as DNA21386.

Figure 88 shows a nucleotide sequence (SEQ ID NO:88) designated herein as DNA22868.

Figure 89 shows a nucleotide sequence (SEQ ID NO:89) designated herein as DNA23694.

Figure 90 shows a nucleotide sequence (SEQ ID NO:90) designated herein as DNA24050.

Figure 91 shows a nucleotide sequence (SEQ ID NO:91) designated herein as DNA24074.

Figure 92 shows a nucleotide sequence (SEQ ID NO:92) designated herein as DNA24787.

Figure 93 shows a nucleotide sequence (SEQ ID NO:93) designated herein as DNA28242.

Figure 94 shows a nucleotide sequence (SEQ ID NO:94) designated herein as DNA28254.

Figure 95 shows a nucleotide sequence (SEQ ID NO:95) designated herein as DNA31751.

Figure 96 shows a nucleotide sequence (SEQ ID NO:96) designated herein as DNA32922.

Figure 97 shows a nucleotide sequence (SEQ ID NO:97) designated herein as DNA33439.

Figure 98 shows a nucleotide sequence (SEQ ID NO:98) designated herein as DNA34508.

Figure 99 shows a nucleotide sequence (SEQ ID NO:99) designated herein as DNA34807.

Figure 100 shows a nucleotide sequence (SEQ ID NO:100) designated herein as DNA34832.

Figure 101 shows a nucleotide sequence (SEQ ID NO:101) designated herein as DNA36223.

Figure 102 shows a nucleotide sequence (SEQ ID NO:102) designated herein as DNA36240.

Figure 103 shows a nucleotide sequence (SEQ ID NO:103) designated herein as DNA36490.

Figure 104 shows a nucleotide sequence (SEQ ID NO:104) designated herein as DNA36516.

Figure 105 shows a nucleotide sequence (SEQ ID NO:105) designated herein as DNA36533.

Figure 106 shows a nucleotide sequence (SEQ ID NO:106) designated herein as DNA36538.

Figure 107 shows a nucleotide sequence (SEQ ID NO:107) designated herein as DNA36788.

Figure 108 shows a nucleotide sequence (SEQ ID NO:108) designated herein as DNA36818.

Figure 109 shows a nucleotide sequence (SEQ ID NO:109) designated herein as DNA36868.

Figure 110 shows a nucleotide sequence (SEQ ID NO:110) designated herein as DNA37393.

Figure 111 shows a nucleotide sequence (SEQ ID NO:111) designated herein as DNA27588.

Figure 112 shows a nucleotide sequence (SEQ ID NO:112) designated herein as DNA37602.

Figure 113 shows a nucleotide sequence (SEQ ID NO:113) designated herein as DNA37642.

Figure 114 shows a nucleotide sequence (SEQ ID NO:114) designated herein as DNA37676.

Figure 115 shows a nucleotide sequence (SEQ ID NO:115) designated herein as DNA37721.

Figure 116 shows a nucleotide sequence (SEQ ID NO:116) designated herein as DNA37759.

Figure 117 shows a nucleotide sequence (SEQ ID NO:117) designated herein as DNA37857.

Figure 118 shows a nucleotide sequence (SEQ ID NO:118) designated herein as DNA37937.

Figure 119 shows a nucleotide sequence (SEQ ID NO:119) designated herein as DNA38037.

Figure 120 shows a nucleotide sequence (SEQ ID NO:120) designated herein as DNA38050.

Figure 121 shows a nucleotide sequence (SEQ ID NO:121) designated herein as DNA38053.

Figure 122 shows a nucleotide sequence (SEQ ID NO:122) designated herein as DNA38312.

Figure 123 shows a nucleotide sequence (SEQ ID NO:123) designated herein as DNA38360.

Figure 124 shows a nucleotide sequence (SEQ ID NO:124) designated herein as DNA38600.

Figure 125 shows a nucleotide sequence (SEQ ID NO:125) designated herein as DNA38720.

Figure 126 shows a nucleotide sequence (SEQ ID NO:126) designated herein as DNA38727.

Figure 127 shows a nucleotide sequence (SEQ ID NO:127) designated herein as DNA38731.

Figure 128 shows a nucleotide sequence (SEQ ID NO:128) designated herein as DNA38810.

Figure 129 shows a nucleotide sequence (SEQ ID NO:129) designated herein as DNA38814.

Figure 130 shows a nucleotide sequence (SEQ ID NO:130) designated herein as DNA39378.

Figure 131 shows a nucleotide sequence (SEQ ID NO:131) designated herein as DNA40050.

Figure 132 shows a nucleotide sequence (SEQ ID NO:132) designated herein as DNA40375.

Figure 133 shows a nucleotide sequence (SEQ ID NO:133) designated herein as DNA40382.

Figure 134 shows a nucleotide sequence (SEQ ID NO:134) designated herein as DNA40394.

Figure 135 shows a nucleotide sequence (SEQ ID NO:135) designated herein as DNA40461.

Figure 136 shows a nucleotide sequence (SEQ ID NO:136) designated herein as DNA40735.

Figure 137 shows a nucleotide sequence (SEQ ID NO:137) designated herein as DNA40736.

Figure 138 shows a nucleotide sequence (SEQ ID NO:138) designated herein as DNA40738.

Figure 139 shows a nucleotide sequence (SEQ ID NO:139) designated herein as DNA40739.

Figure 140 shows a nucleotide sequence (SEQ ID NO:140) designated herein as DNA41144.

Figure 141 shows a nucleotide sequence (SEQ ID NO:141) designated herein as DNA41161.

Figure 142 shows a nucleotide sequence (SEQ ID NO:142) designated herein as DNA41186.

Figure 143 shows a nucleotide sequence (SEQ ID NO:143) designated herein as DNA41250.

Figure 144 shows a nucleotide sequence (SEQ ID NO:144) designated herein as DNA41284.

Figure 145 shows a nucleotide sequence (SEQ ID NO:145) designated herein as DNA41303.

Figure 146 shows a nucleotide sequence (SEQ ID NO:146) designated herein as DNA41326.

Figure 147 shows a nucleotide sequence (SEQ ID NO:147) designated herein as DNA41444.

Figure 148 shows a nucleotide sequence (SEQ ID NO:148) designated herein as DNA41445.

Figure 149 shows a nucleotide sequence (SEQ ID NO:149) designated herein as DNA41452.

Figure 150 shows a nucleotide sequence (SEQ ID NO:150) designated herein as DNA41456.

Figure 151 shows a nucleotide sequence (SEQ ID NO:151) designated herein as DNA41458.

Figure 152 shows a nucleotide sequence (SEQ ID NO:152) designated herein as DNA41462.

Figure 153 shows a nucleotide sequence (SEQ ID NO:153) designated herein as DNA41465.

Figure 154 shows a nucleotide sequence (SEQ ID NO:154) designated herein as DNA41475.

Figure 155 shows a nucleotide sequence (SEQ ID NO:155) designated herein as DNA41514.

Figure 156 shows a nucleotide sequence (SEQ ID NO:156) designated herein as DNA41565.

Figure 157 shows a nucleotide sequence (SEQ ID NO:157) designated herein as DNA41566.

Figure 158 shows a nucleotide sequence (SEQ ID NO:158) designated herein as DNA41626.

Figure 159 shows a nucleotide sequence (SEQ ID NO:159) designated herein as DNA41709.

Figure 160 shows a nucleotide sequence (SEQ ID NO:160) designated herein as DNA41775.

Figure 161 shows a nucleotide sequence (SEQ ID NO:161) designated herein as DNA41784.

Figure 162 shows a nucleotide sequence (SEQ ID NO:162) designated herein as DNA42194.

Figure 163 shows a nucleotide sequence (SEQ ID NO:163) designated herein as DNA42279.

Figure 164 shows a nucleotide sequence (SEQ ID NO:164) designated herein as DNA42314.

Figure 165 shows a nucleotide sequence (SEQ ID NO:165) designated herein as DNA42331.

Figure 166 shows a nucleotide sequence (SEQ ID NO:166) designated herein as DNA42358.

Figure 167 shows a nucleotide sequence (SEQ ID NO:167) designated herein as DNA42858.

Figure 168 shows a nucleotide sequence (SEQ ID NO:168) designated herein as DNA42870.

Figure 169 shows a nucleotide sequence (SEQ ID NO:169) designated herein as DNA42875.

Figure 170 shows a nucleotide sequence (SEQ ID NO:170) designated herein as DNA43197.

Figure 171 shows a nucleotide sequence (SEQ ID NO:171) designated herein as DNA43203.

Figure 172 shows a nucleotide sequence (SEQ ID NO:172) designated herein as DNA43295.

Figure 173 shows a nucleotide sequence (SEQ ID NO:173) designated herein as DNA43301.

Figure 174 shows a nucleotide sequence (SEQ ID NO:174) designated herein as DNA43363.

Figure 175 shows a nucleotide sequence (SEQ ID NO:175) designated herein as DNA43420.

Figure 176 shows a nucleotide sequence (SEQ ID NO:176) designated herein as DNA443479.

Figure 177 shows a nucleotide sequence (SEQ ID NO:177) designated herein as DNA43489.

Figure 178 shows a nucleotide sequence (SEQ ID NO:178) designated herein as DNA43498.

Figure 179 shows a nucleotide sequence (SEQ ID NO:179) designated herein as DNA43509.

Figure 180 shows a nucleotide sequence (SEQ ID NO:180) designated herein as DNA43512.

Figure 181 shows a nucleotide sequence (SEQ ID NO:181) designated herein as DNA43531.

Figure 182 shows a nucleotide sequence (SEQ ID NO:182) designated herein as DNA43546.

Figure 183 shows a nucleotide sequence (SEQ ID NO:183) designated herein as DNA43586.

Figure 184 shows a nucleotide sequence (SEQ ID NO:184) designated herein as DNA43862.

Figure 185 shows a nucleotide sequence (SEQ ID NO:185) designated herein as DNA43887.

Figure 186 shows a nucleotide sequence (SEQ ID NO:186) designated herein as DNA43936.

Figure 187 shows a nucleotide sequence (SEQ ID NO:187) designated herein as DNA43961.

Figure 188 shows a nucleotide sequence (SEQ ID NO:188) designated herein as DNA43971.

Figure 189 shows a nucleotide sequence (SEQ ID NO:189) designated herein as DNA44048.

Figure 190 shows a nucleotide sequence (SEQ ID NO:190) designated herein as DNA44920.

Figure 191 shows a nucleotide sequence (SEQ ID NO:191) designated herein as DNA44922.

Figure 192 shows a nucleotide sequence (SEQ ID NO:192) designated herein as DNA44934.

Figure 193 shows a nucleotide sequence (SEQ ID NO:193) designated herein as DNA44987.

Figure 194 shows a nucleotide sequence (SEQ ID NO:194) designated herein as DNA45014.

Figure 195 shows a nucleotide sequence (SEQ ID NO:195) designated herein as DNA45030.

Figure 196 shows a nucleotide sequence (SEQ ID NO:196) designated herein as DNA45051.

Figure 197 shows a nucleotide sequence (SEQ ID NO:197) designated herein as DNA45064.

Figure 198 shows a nucleotide sequence (SEQ ID NO:198) designated herein as DNA45282.

Figure 199 shows a nucleotide sequence (SEQ ID NO:199) designated herein as DNA45288.

Figure 200 shows a nucleotide sequence (SEQ ID NO:200) designated herein as DNA45300.

Figure 201 shows a nucleotide sequence (SEQ ID NO:201) designated herein as DNA45740.

Figure 202 shows a nucleotide sequence (SEQ ID NO:202) designated herein as DNA45759.

Figure 203 shows a nucleotide sequence (SEQ ID NO:203) designated herein as DNA45784.

Figure 204 shows a nucleotide sequence (SEQ ID NO:204) designated herein as DNA45789.

Figure 205 shows a nucleotide sequence (SEQ ID NO:205) designated herein as DNA45816.

Figure 206 shows a nucleotide sequence (SEQ ID NO:206) designated herein as DNA45944.

Figure 207 shows a nucleotide sequence (SEQ ID NO:207) designated herein as DNA45954.

Figure 208 shows a nucleotide sequence (SEQ ID NO:208) designated herein as DNA45964.

Figure 209 shows a nucleotide sequence (SEQ ID NO:209) designated herein as DNA45993.

Figure 210 shows a nucleotide sequence (SEQ ID NO:210) designated herein as DNA46092.

Figure 211 shows a nucleotide sequence (SEQ ID NO:211) designated herein as DNA46213.

Figure 212 shows a nucleotide sequence (SEQ ID NO:212) designated herein as DNA46215.

Figure 213 shows a nucleotide sequence (SEQ ID NO:213) designated herein as DNA46226.

Figure 214 shows a nucleotide sequence (SEQ ID NO:214) designated herein as DNA46328.

Figure 215 shows a nucleotide sequence (SEQ ID NO:215) designated herein as DNA47580.

Figure 216 shows a nucleotide sequence (SEQ ID NO:216) designated herein as DNA47691.

Figure 217 shows a nucleotide sequence (SEQ ID NO:217) designated herein as DNA47751.

Figure 218 shows a nucleotide sequence (SEQ ID NO:218) designated herein as DNA47835.

Figure 219 shows a nucleotide sequence (SEQ ID NO:219) designated herein as DNA47858.

Figure 220 shows a nucleotide sequence (SEQ ID NO:220) designated herein as DNA47890.

Figure 221 shows a nucleotide sequence (SEQ ID NO:221) designated herein as DNA47930.

Figure 222 shows a nucleotide sequence (SEQ ID NO:222) designated herein as DNA47990.

Figure 223 shows a nucleotide sequence (SEQ ID NO:223) designated herein as DNA48054.

Figure 224 shows a nucleotide sequence (SEQ ID NO:224) designated herein as DNA48124.

Figure 225 shows a nucleotide sequence (SEQ ID NO:225) designated herein as DNA48131.

Figure 226 shows a nucleotide sequence (SEQ ID NO:226) designated herein as DNA48162.

Figure 227 shows a nucleotide sequence (SEQ ID NO:227) designated herein as DNA48209.

Figure 228 shows a nucleotide sequence (SEQ ID NO:228) designated herein as DNA48389.

Figure 229 shows a nucleotide sequence (SEQ ID NO:229) designated herein as DNA48446.

Figure 230 shows a nucleotide sequence (SEQ ID NO:230) designated herein as DNA48466.

Figure 231 shows a nucleotide sequence (SEQ ID NO:231) designated herein as DNA48576.

Figure 232 shows a nucleotide sequence (SEQ ID NO:232) designated herein as DNA48598.

Figure 233 shows a nucleotide sequence (SEQ ID NO:233) designated herein as DNA48666.

Figure 234 shows a nucleotide sequence (SEQ ID NO:234) designated herein as DNA48748.

Figure 235 shows a nucleotide sequence (SEQ ID NO:235) designated herein as DNA48777.

Figure 236 shows a nucleotide sequence (SEQ ID NO:236) designated herein as DNA48830.

Figure 237 shows a nucleotide sequence (SEQ ID NO:237) designated herein as DNA49352.

Figure 238 shows a nucleotide sequence (SEQ ID NO:238) designated herein as DNA49407.

Figure 239 shows a nucleotide sequence (SEQ ID NO:239) designated herein as DNA49448.

Figure 240 shows a nucleotide sequence (SEQ ID NO:240) designated herein as DNA49528.

Figure 241 shows a nucleotide sequence (SEQ ID NO:241) designated herein as DNA49529.

Figure 242 shows a nucleotide sequence (SEQ ID NO:242) designated herein as DNA49948.

Figure 243 shows a nucleotide sequence (SEQ ID NO:243) designated herein as DNA49956.

Figure 244 shows a nucleotide sequence (SEQ ID NO:244) designated herein as DNA49992.

Figure 245 shows a nucleotide sequence (SEQ ID NO:245) designated herein as DNA50307.

Figure 246 shows a nucleotide sequence (SEQ ID NO:246) designated herein as DNA50319.

Figure 247 shows a nucleotide sequence (SEQ ID NO:247) designated herein as DNA50346.

Figure 248 shows a nucleotide sequence (SEQ ID NO:248) designated herein as DNA50354.

Figure 249 shows a nucleotide sequence (SEQ ID NO:249) designated herein as DNA50356.

Figure 250 shows a nucleotide sequence (SEQ ID NO:250) designated herein as DNA50405.

Figure 251 shows a nucleotide sequence (SEQ ID NO:251) designated herein as DNA50421.

Figure 252 shows a nucleotide sequence (SEQ ID NO:252) designated herein as DNA50423.

Figure 253 shows a nucleotide sequence (SEQ ID NO:253) designated herein as DNA50527.

Figure 254 shows a nucleotide sequence (SEQ ID NO:254) designated herein as DNA50584.

Figure 255 shows a nucleotide sequence (SEQ ID NO:255) designated herein as DNA50626.

Figure 256 shows a nucleotide sequence (SEQ ID NO:256) designated herein as DNA50637.

Figure 257 shows a nucleotide sequence (SEQ ID NO:257) designated herein as DNA50650.

Figure 258 shows a nucleotide sequence (SEQ ID NO:258) designated herein as DNA50674.

Figure 259 shows a nucleotide sequence (SEQ ID NO:259) designated herein as DNA50675.

Figure 260 shows a nucleotide sequence (SEQ ID NO:260) designated herein as DNA50698.

Figure 261 shows a nucleotide sequence (SEQ ID NO:261) designated herein as DNA50730.

Figure 262 shows a nucleotide sequence (SEQ ID NO:262) designated herein as DNA50737.

Figure 263 shows a nucleotide sequence (SEQ ID NO:263) designated herein as DNA51003.

Figure 264 shows a nucleotide sequence (SEQ ID NO:264) designated herein as DNA51010.

Figure 265 shows a nucleotide sequence (SEQ ID NO:265) designated herein as DNA51059.

Figure 266 shows a nucleotide sequence (SEQ ID NO:266) designated herein as DNA51413.

Figure 267 shows a nucleotide sequence (SEQ ID NO:267) designated herein as DNA51712.

Figure 268 shows a nucleotide sequence (SEQ ID NO:268) designated herein as DNA51795.

Figure 269 shows a nucleotide sequence (SEQ ID NO:269) designated herein as DNA52199.

Figure 270 shows a nucleotide sequence (SEQ ID NO:270) designated herein as DNA52218.

Figure 271 shows a nucleotide sequence (SEQ ID NO:271) designated herein as DNA52352.

Figure 272 shows a nucleotide sequence (SEQ ID NO:272) designated herein as DNA54446.

Figure 273 shows a nucleotide sequence (SEQ ID NO:273) designated herein as DNA54552.

Figure 274 shows a nucleotide sequence (SEQ ID NO:274) designated herein as DNA54580.

Figure 275 shows a nucleotide sequence (SEQ ID NO:275) designated herein as DNA54623.

Figure 276 shows a nucleotide sequence (SEQ ID NO:276) designated herein as DNA54672.

Figure 277 shows a nucleotide sequence (SEQ ID NO:277) designated herein as DNA54840.

Figure 278 shows a nucleotide sequence (SEQ ID NO:278) designated herein as DNA54856.

Figure 279 shows a nucleotide sequence (SEQ ID NO:279) designated herein as DNA54882.

Figure 280 shows a nucleotide sequence (SEQ ID NO:280) designated herein as DNA54943.

Figure 281 shows a nucleotide sequence (SEQ ID NO:281) designated herein as DNA54970.

Figure 282 shows a nucleotide sequence (SEQ ID NO:282) designated herein as DNA55134.

Figure 283 shows a nucleotide sequence (SEQ ID NO:283) designated herein as DNA55198.

Figure 284 shows a nucleotide sequence (SEQ ID NO:284) designated herein as DNA55199.

Figure 285 shows a nucleotide sequence (SEQ ID NO:285) designated herein as DNA55292.

Figure 286 shows a nucleotide sequence (SEQ ID NO:286) designated herein as DNA55646.

Figure 287 shows a nucleotide sequence (SEQ ID NO:287) designated herein as DNA56553.

Figure 288 shows a nucleotide sequence (SEQ ID NO:288) designated herein as DNA56554.

Figure 289 shows a nucleotide sequence (SEQ ID NO:289) designated herein as DNA56556.

Figure 290 shows a nucleotide sequence (SEQ ID NO:290) designated herein as DNA56587.

Figure 291 shows a nucleotide sequence (SEQ ID NO:291) designated herein as DNA56590.

Figure 292 shows a nucleotide sequence (SEQ ID NO:292) designated herein as DNA56600.

Figure 293 shows a nucleotide sequence (SEQ ID NO:293) designated herein as DNA56648.

Figure 294 shows a nucleotide sequence (SEQ ID NO:294) designated herein as DNA56650.

Figure 295 shows a nucleotide sequence (SEQ ID NO:295) designated herein as DNA56707.

Figure 296 shows a nucleotide sequence (SEQ ID NO:296) designated herein as DNA56717.

Figure 297 shows a nucleotide sequence (SEQ ID NO:297) designated herein as DNA58387.

Figure 298 shows a nucleotide sequence (SEQ ID NO:298) designated herein as DNA58414.

Figure 299 shows a nucleotide sequence (SEQ ID NO:299) designated herein as DNA58529.

Figure 300 shows a nucleotide sequence (SEQ ID NO:300) designated herein as DNA59385.

Figure 301 shows a nucleotide sequence (SEQ ID NO:301) designated herein as DNA59789.

Figure 302 shows a nucleotide sequence (SEQ ID NO:302) designated herein as DNA60321.

Figure 303 shows a nucleotide sequence (SEQ ID NO:303) designated herein as DNA60370.

Figure 304 shows a nucleotide sequence (SEQ ID NO:304) designated herein as DNA60406.

Figure 305 shows a nucleotide sequence (SEQ ID NO:305) designated herein as DNA60438.

Figure 306 shows a nucleotide sequence (SEQ ID NO:306) designated herein as DNA60460.

Figure 307 shows a nucleotide sequence (SEQ ID NO:307) designated herein as DNA60466.

Figure 308 shows a nucleotide sequence (SEQ ID NO:308) designated herein as DNA60508.

Figure 309 shows a nucleotide sequence (SEQ ID NO:309) designated herein as DNA60542.

Figure 310 shows a nucleotide sequence (SEQ ID NO:310) designated herein as DNA60590.

Figure 311 shows a nucleotide sequence (SEQ ID NO:311) designated herein as DNA61350.

Figure 312 shows a nucleotide sequence (SEQ ID NO:312) designated herein as DNA61356.

Figure 313 shows a nucleotide sequence (SEQ ID NO:313) designated herein as DNA61478.

Figure 314 shows a nucleotide sequence (SEQ ID NO:314) designated herein as DNA61513.

Figure 315 shows a nucleotide sequence (SEQ ID NO:315) designated herein as DNA61561.

Figure 316 shows a nucleotide sequence (SEQ ID NO:316) designated herein as DNA61895.

Figure 317 shows a nucleotide sequence (SEQ ID NO:317) designated herein as DNA61930.

Figure 318 shows a nucleotide sequence (SEQ ID NO:318) designated herein as DNA61953.

Figure 319 shows a nucleotide sequence (SEQ ID NO:319) designated herein as DNA62011.

Figure 320 shows a nucleotide sequence (SEQ ID NO:320) designated herein as DNA62080.

Figure 321 shows a nucleotide sequence (SEQ ID NO:321) designated herein as DNA62126.

Figure 322 shows a nucleotide sequence (SEQ ID NO:322) designated herein as DNA62154.

Figure 323 shows a nucleotide sequence (SEQ ID NO:323) designated herein as DNA62170.

Figure 324 shows a nucleotide sequence (SEQ ID NO:324) designated herein as DNA62193.

Figure 325 shows a nucleotide sequence (SEQ ID NO:325) designated herein as DNA62261.

Figure 326 shows a nucleotide sequence (SEQ ID NO:326) designated herein as DNA62291.

Figure 327 shows a nucleotide sequence (SEQ ID NO:327) designated herein as DNA62422.

Figure 328 shows a nucleotide sequence (SEQ ID NO:328) designated herein as DNA62436.

Figure 329 shows a nucleotide sequence (SEQ ID NO:329) designated herein as DNA62524.

Figure 330 shows a nucleotide sequence (SEQ ID NO:330) designated herein as DNA62589.

Figure 331 shows a nucleotide sequence (SEQ ID NO:331) designated herein as DNA63878.

Figure 332 shows a nucleotide sequence (SEQ ID NO:332) designated herein as DNA64017.

Figure 333 shows a nucleotide sequence (SEQ ID NO:333) designated herein as DNA64045.

Figure 334 shows a nucleotide sequence (SEQ ID NO:334) designated herein as DNA64101.

Figure 335 shows a nucleotide sequence (SEQ ID NO:335) designated herein as DNA64183.

Figure 336 shows a nucleotide sequence (SEQ ID NO:336) designated herein as DNA64193.

Figure 337 shows a nucleotide sequence (SEQ ID NO:337) designated herein as DNA64199.

Figure 338 shows a nucleotide sequence (SEQ ID NO:338) designated herein as DNA64268.

Figure 339 shows a nucleotide sequence (SEQ ID NO:339) designated herein as DNA64304.

Figure 340 shows a nucleotide sequence (SEQ ID NO:340) designated herein as DNA64453.

Figure 341 shows a nucleotide sequence (SEQ ID NO:341) designated herein as DNA64458.

Figure 342 shows a nucleotide sequence (SEQ ID NO:342) designated herein as DNA64512.

Figure 343 shows a nucleotide sequence (SEQ ID NO:343) designated herein as DNA64540.

Figure 344 shows a nucleotide sequence (SEQ ID NO:344) designated herein as DNA64552.

Figure 345 shows a nucleotide sequence (SEQ ID NO:345) designated herein as DNA64557.

Figure 346 shows a nucleotide sequence (SEQ ID NO:346) designated herein as DNA64569.

Figure 347 shows a nucleotide sequence (SEQ ID NO:347) designated herein as DNA64627.

Figure 348 shows a nucleotide sequence (SEQ ID NO:348) designated herein as DNA64745.

Figure 349 shows a nucleotide sequence (SEQ ID NO:349) designated herein as DNA64784.

Figure 350 shows a nucleotide sequence (SEQ ID NO:350) designated herein as DNA65609.

Figure 351 shows a nucleotide sequence (SEQ ID NO:351) designated herein as DNA65644.

Figure 352 shows a nucleotide sequence (SEQ ID NO:352) designated herein as DNA65720.

Figure 353 shows a nucleotide sequence (SEQ ID NO:353) designated herein as DNA65752.

Figure 354 shows a nucleotide sequence (SEQ ID NO:354) designated herein as DNA65771.

Figure 355 shows a nucleotide sequence (SEQ ID NO:355) designated herein as DNA65833.

Figure 356 shows a nucleotide sequence (SEQ ID NO:356) designated herein as DNA65836.

Figure 357 shows a nucleotide sequence (SEQ ID NO:357) designated herein as DNA65864.

Figure 358 shows a nucleotide sequence (SEQ ID NO:358) designated herein as DNA65869.

Figure 359 shows a nucleotide sequence (SEQ ID NO:359) designated herein as DNA65928.

Figure 360 shows a nucleotide sequence (SEQ ID NO:360) designated herein as DNA66065.

Figure 361 shows a nucleotide sequence (SEQ ID NO:361) designated herein as DNA66095.

Figure 362 shows a nucleotide sequence (SEQ ID NO:362) designated herein as DNA66197.

Figure 363 shows a nucleotide sequence (SEQ ID NO:363) designated herein as DNA66217.

Figure 364 shows a nucleotide sequence (SEQ ID NO:364) designated herein as DNA66231.

Figure 365 shows a nucleotide sequence (SEQ ID NO:365) designated herein as DNA66404.

Figure 366 shows a nucleotide sequence (SEQ ID NO:366) designated herein as DNA66432.

Figure 367 shows a nucleotide sequence (SEQ ID NO:367) designated herein as DNA67076.

Figure 368 shows a nucleotide sequence (SEQ ID NO:368) designated herein as DNA68013.

Figure 369 shows a nucleotide sequence (SEQ ID NO:369) designated herein as DNA68018.

Figure 370 shows a nucleotide sequence (SEQ ID NO:370) designated herein as DNA68034.

Figure 371 shows a nucleotide sequence (SEQ ID NO:371) designated herein as DNA68119.

Figure 372 shows a nucleotide sequence (SEQ ID NO:372) designated herein as DNA68248.

Figure 373 shows a nucleotide sequence (SEQ ID NO:373) designated herein as DNA68383.

Figure 374 shows a nucleotide sequence (SEQ ID NO:374) designated herein as DNA68423.

Figure 375 shows a nucleotide sequence (SEQ ID NO:375) designated herein as DNA68441.

Figure 376 shows a nucleotide sequence (SEQ ID NO:376) designated herein as DNA68459.

Figure 377 shows a nucleotide sequence (SEQ ID NO:377) designated herein as DNA68509.

Figure 378 shows a nucleotide sequence (SEQ ID NO:378) designated herein as DNA68514.

Figure 379 shows a nucleotide sequence (SEQ ID NO:379) designated herein as DNA68521.

Figure 380 shows a nucleotide sequence (SEQ ID NO:380) designated herein as DNA68532.

Figure 381 shows a nucleotide sequence (SEQ ID NO:381) designated herein as DNA68540.

Figure 382 shows a nucleotide sequence (SEQ ID NO:382) designated herein as DNA68561.

Figure 383 shows a nucleotide sequence (SEQ ID NO:383) designated herein as DNA68585.

Figure 384 shows a nucleotide sequence (SEQ ID NO:384) designated herein as DNA69491.

Figure 385 shows a nucleotide sequence (SEQ ID NO:385) designated herein as DNA70222.

Figure 386 shows a nucleotide sequence (SEQ ID NO:386) designated herein as DNA70239.

Figure 387 shows a nucleotide sequence (SEQ ID NO:387) designated herein as DNA70244.

Figure 388 shows a nucleotide sequence (SEQ ID NO:388) designated herein as DNA70349.

Figure 389 shows a nucleotide sequence (SEQ ID NO:389) designated herein as DNA70400.

Figure 390 shows a nucleotide sequence (SEQ ID NO:390) designated herein as DNA70413.

Figure 391 shows a nucleotide sequence (SEQ ID NO:391) designated herein as DNA70526.

Figure 392 shows a nucleotide sequence (SEQ ID NO:392) designated herein as DNA70685.

Figure 393 shows a nucleotide sequence (SEQ ID NO:393) designated herein as DNA70732.

Figure 394 shows a nucleotide sequence (SEQ ID NO:394) designated herein as DNA72634.

Figure 395 shows a nucleotide sequence (SEQ ID NO:395) designated herein as DNA72683.

Figure 396 shows a nucleotide sequence (SEQ ID NO:396) designated herein as DNA72695.

Figure 397 shows a nucleotide sequence (SEQ ID NO:397) designated herein as DNA72864.

Figure 398 shows a nucleotide sequence (SEQ ID NO:398) designated herein as DNA73156.

Figure 399 shows a nucleotide sequence (SEQ ID NO:399) designated herein as DNA73275.

Figure 400 shows a nucleotide sequence (SEQ ID NO:400) designated herein as DNA74052.

Figure 401 shows a nucleotide sequence (SEQ ID NO:401) designated herein as DNA74063.

Figure 402 shows a nucleotide sequence (SEQ ID NO:402) designated herein as DNA74072.

Figure 403 shows a nucleotide sequence (SEQ ID NO:403) designated herein as DNA74140.

Figure 404 shows a nucleotide sequence (SEQ ID NO:404) designated herein as DNA74216.

Figure 405 shows a nucleotide sequence (SEQ ID NO:405) designated herein as DNA74218.

Figure 406 shows a nucleotide sequence (SEQ ID NO:406) designated herein as DNA74228.

Figure 407 shows a nucleotide sequence (SEQ ID NO:407) designated herein as DNA74256.

Figure 408 shows a nucleotide sequence (SEQ ID NO:408) designated herein as DNA75062.

Figure 409 shows a nucleotide sequence (SEQ ID NO:409) designated herein as DNA76137.

Figure 410 shows a nucleotide sequence (SEQ ID NO:410) designated herein as DNA76158.

Figure 411 shows a nucleotide sequence (SEQ ID NO:411) designated herein as DNA77098.

Figure 412 shows a nucleotide sequence (SEQ ID NO:412) designated herein as DNA77791.

Figure 413 shows a nucleotide sequence (SEQ ID NO:413) designated herein as DNA77968.

Figure 414 shows a nucleotide sequence (SEQ ID NO:414) designated herein as DNA77976.

Figure 415 shows a nucleotide sequence (SEQ ID NO:415) designated herein as DNA78017.

Figure 416 shows a nucleotide sequence (SEQ ID NO:416) designated herein as DNA78095.

Figure 417 shows a nucleotide sequence (SEQ ID NO:417) designated herein as DNA78103.

Figure 418 shows a nucleotide sequence (SEQ ID NO:418) designated herein as DNA78113.

Figure 419 shows a nucleotide sequence (SEQ ID NO:419) designated herein as DNA78746.

Figure 420 shows a nucleotide sequence (SEQ ID NO:420) designated herein as DNA78759.

Figure 421 shows a nucleotide sequence (SEQ ID NO:421) designated herein as DNA78796.

Figure 422 shows a nucleotide sequence (SEQ ID NO:422) designated herein as DNA79561.

Figure 423 shows a nucleotide sequence (SEQ ID NO:423) designated herein as DNA79602.

Figure 424 shows a nucleotide sequence (SEQ ID NO:424) designated herein as DNA79617.

Figure 425 shows a nucleotide sequence (SEQ ID NO:425) designated herein as DNA79628.

Figure 426 shows a nucleotide sequence (SEQ ID NO:426) designated herein as DNA79640.

Figure 427 shows a nucleotide sequence (SEQ ID NO:427) designated herein as DNA79661.

Figure 428 shows a nucleotide sequence (SEQ ID NO:428) designated herein as DNA79684.

Figure 429 shows a nucleotide sequence (SEQ ID NO:429) designated herein as DNA79717.

Figure 430 shows a nucleotide sequence (SEQ ID NO:430) designated herein as DNA79733.

Figure 431 shows a nucleotide sequence (SEQ ID NO:431) designated herein as DNA79970.

Figure 432 shows a nucleotide sequence (SEQ ID NO:432) designated herein as DNA80050.

Figure 433 shows a nucleotide sequence (SEQ ID NO:433) designated herein as DNA80247.

Figure 434 shows a nucleotide sequence (SEQ ID NO:434) designated herein as DNA80265.

Figure 435 shows a nucleotide sequence (SEQ ID NO:435) designated herein as DNA80615.

Figure 436 shows a nucleotide sequence (SEQ ID NO:436) designated herein as DNA80623.

Figure 437 shows a nucleotide sequence (SEQ ID NO:437) designated herein as DNA80627.

Figure 438 shows a nucleotide sequence (SEQ ID NO:438) designated herein as DNA81896.

Figure 439 shows a nucleotide sequence (SEQ ID NO:439) designated herein as DNA81918.

Figure 440 shows a nucleotide sequence (SEQ ID NO:440) designated herein as DNA81976.

Figure 441 shows a nucleotide sequence (SEQ ID NO:441) designated herein as DNA82017.

Figure 442 shows a nucleotide sequence (SEQ ID NO:442) designated herein as DNA82024.

Figure 443 shows a nucleotide sequence (SEQ ID NO:443) designated herein as DNA82027.

Figure 444 shows a nucleotide sequence (SEQ ID NO:444) designated herein as DNA82115.

Figure 445 shows a nucleotide sequence (SEQ ID NO:445) designated herein as DNA82154.

Figure 446 shows a nucleotide sequence (SEQ ID NO:446) designated herein as DNA82157.

Figure 447 shows a nucleotide sequence (SEQ ID NO:447) designated herein as DNA82166.

Figure 448 shows a nucleotide sequence (SEQ ID NO:448) designated herein as DNA82182.

Figure 449 shows a nucleotide sequence (SEQ ID NO:449) designated herein as DNA82212.

Figure 450 shows a nucleotide sequence (SEQ ID NO:450) designated herein as DNA82498.

Figure 451 shows a nucleotide sequence (SEQ ID NO:451) designated herein as DNA82499.

Figure 452 shows a nucleotide sequence (SEQ ID NO:452) designated herein as DNA82504.

Figure 453 shows a nucleotide sequence (SEQ ID NO:453) designated herein as DNA82531.

Figure 454 shows a nucleotide sequence (SEQ ID NO:454) designated herein as DNA82693.

Figure 455 shows a nucleotide sequence (SEQ ID NO:455) designated herein as DNA82702.

Figure 456 shows a nucleotide sequence (SEQ ID NO:456) designated herein as DNA82786.

Figure 457 shows a nucleotide sequence (SEQ ID NO:457) designated herein as DNA82851.

Figure 458 shows a nucleotide sequence (SEQ ID NO:458) designated herein as DNA82898.

Figure 459 shows a nucleotide sequence (SEQ ID NO:459) designated herein as DNA82935.

Figure 460 shows a nucleotide sequence (SEQ ID NO:460) designated herein as DNA82977.

Figure 461 shows a nucleotide sequence (SEQ ID NO:461) designated herein as DNA82989.

Figure 462 shows a nucleotide sequence (SEQ ID NO:462) designated herein as DNA83628.

Figure 463 shows a nucleotide sequence (SEQ ID NO:463) designated herein as DNA83630.

Figure 464 shows a nucleotide sequence (SEQ ID NO:464) designated herein as DNA83749.

Figure 465 shows a nucleotide sequence (SEQ ID NO:465) designated herein as DNA83772.

Figure 466 shows a nucleotide sequence (SEQ ID NO:466) designated herein as DNA83800.

Figure 467 shows a nucleotide sequence (SEQ ID NO:467) designated herein as DNA83950.

Figure 468 shows a nucleotide sequence (SEQ ID NO:468) designated herein as DNA84027.

Figure 469 shows a nucleotide sequence (SEQ ID NO:469) designated herein as DNA84076.

Figure 470 shows a nucleotide sequence (SEQ ID NO:470) designated herein as DNA84109.

Figure 471 shows a nucleotide sequence (SEQ ID NO:471) designated herein as DNA85072.

Figure 472 shows a nucleotide sequence (SEQ ID NO:472) designated herein as DNA85154.

Figure 473 shows a nucleotide sequence (SEQ ID NO:473) designated herein as DNA85193.

Figure 474 shows a nucleotide sequence (SEQ ID NO:474) designated herein as DNA85224.

Figure 475 shows a nucleotide sequence (SEQ ID NO:475) designated herein as DNA85237.

Figure 476 shows a nucleotide sequence (SEQ ID NO:476) designated herein as DNA85289.

Figure 477 shows a nucleotide sequence (SEQ ID NO:477) designated herein as DNA85357.

Figure 478 shows a nucleotide sequence (SEQ ID NO:478) designated herein as DNA85361.

Figure 479 shows a nucleotide sequence (SEQ ID NO:479) designated herein as DNA85371.

Figure 480 shows a nucleotide sequence (SEQ ID NO:480) designated herein as DNA86875.

Figure 481 shows a nucleotide sequence (SEQ ID NO:481) designated herein as DNA86876.

Figure 482 shows a nucleotide sequence (SEQ ID NO:482) designated herein as DNA86905.

Figure 483 shows a nucleotide sequence (SEQ ID NO:483) designated herein as DNA86945.

Figure 484 shows a nucleotide sequence (SEQ ID NO:484) designated herein as DNA86969.

Figure 485 shows a nucleotide sequence (SEQ ID NO:485) designated herein as DNA87050.

Figure 486 shows a nucleotide sequence (SEQ ID NO:486) designated herein as DNA87094.

Figure 487 shows a nucleotide sequence (SEQ ID NO:487) designated herein as DNA87126.

Figure 488 shows a nucleotide sequence (SEQ ID NO:488) designated herein as DNA87493.

Figure 489 shows a nucleotide sequence (SEQ ID NO:489) designated herein as DNA87494.

Figure 490 shows a nucleotide sequence (SEQ ID NO:490) designated herein as DNA87505.

Figure 491 shows a nucleotide sequence (SEQ ID NO:491) designated herein as DNA87566.

Figure 492 shows a nucleotide sequence (SEQ ID NO:492) designated herein as DNA87586.

Figure 493 shows a nucleotide sequence (SEQ ID NO:493) designated herein as DNA87649.

Figure 494 shows a nucleotide sequence (SEQ ID NO:494) designated herein as DNA89340.

Figure 495 shows a nucleotide sequence (SEQ ID NO:495) designated herein as DNA89355.

Figure 496 shows a nucleotide sequence (SEQ ID NO:496) designated herein as DNA89365.

Figure 497 shows a nucleotide sequence (SEQ ID NO:497) designated herein as DNA89419.

Figure 498 shows a nucleotide sequence (SEQ ID NO:498) designated herein as DNA89470.

Figure 499 shows a nucleotide sequence (SEQ ID NO:499) designated herein as DNA89480.

Figure 500 shows a nucleotide sequence (SEQ ID NO:500) designated herein as DNA89549.

Figure 501 shows a nucleotide sequence (SEQ ID NO:501) designated herein as DNA89606.

Figure 502 shows a nucleotide sequence (SEQ ID NO:502) designated herein as DNA89615.

Figure 503 shows a nucleotide sequence (SEQ ID NO:503) designated herein as DNA89669.

Figure 504 shows a nucleotide sequence (SEQ ID NO:504) designated herein as DNA89760.

Figure 505 shows a nucleotide sequence (SEQ ID NO:505) designated herein as DNA89766.

Figure 506 shows a nucleotide sequence (SEQ ID NO:506) designated herein as DNA89772.

Figure 507 shows a nucleotide sequence (SEQ ID NO:507) designated herein as DNA89773.

Figure 508 shows a nucleotide sequence (SEQ ID NO:508) designated herein as DNA89774.

Figure 509 shows a nucleotide sequence (SEQ ID NO:509) designated herein as DNA89872.

Figure 510 shows a nucleotide sequence (SEQ ID NO:510) designated herein as DNA89918.

Figure 511 shows a nucleotide sequence (SEQ ID NO:511) designated herein as DNA89928.

Figure 512 shows a nucleotide sequence (SEQ ID NO:512) designated herein as DNA89930.

Figure 513 shows a nucleotide sequence (SEQ ID NO:513) designated herein as DNA91463.

Figure 514 shows a nucleotide sequence (SEQ ID NO:514) designated herein as DNA91507.

Figure 515 shows a nucleotide sequence (SEQ ID NO:515) designated herein as DNA93615.

Figure 516 shows a nucleotide sequence (SEQ ID NO:516) designated herein as DNA94011.

Figure 517 shows a nucleotide sequence (SEQ ID NO:517) designated herein as DNA94043.

Figure 518 shows a nucleotide sequence (SEQ ID NO:518) designated herein as DNA94050.

Figure 519 shows a nucleotide sequence (SEQ ID NO:519) designated herein as DNA94097.

Figure 520 shows a nucleotide sequence (SEQ ID NO:520) designated herein as DNA94098.

Figure 521 shows a nucleotide sequence (SEQ ID NO:521) designated herein as DNA94100.

Figure 522 shows a nucleotide sequence (SEQ ID NO:522) designated herein as DNA94126.

Figure 523 shows a nucleotide sequence (SEQ ID NO:523) designated herein as DNA94136.

Figure 524 shows a nucleotide sequence (SEQ ID NO:524) designated herein as DNA94156.

Figure 525 shows a nucleotide sequence (SEQ ID NO:525) designated herein as DNA94219.

Figure 526 shows a nucleotide sequence (SEQ ID NO:526) designated herein as DNA94254.

Figure 527 shows a nucleotide sequence (SEQ ID NO:527) designated herein as DNA94274.

Figure 528 shows a nucleotide sequence (SEQ ID NO:528) designated herein as DNA94292.

Figure 529 shows a nucleotide sequence (SEQ ID NO:529) designated herein as DNA94360.

Figure 530 shows a nucleotide sequence (SEQ ID NO:530) designated herein as DNA94377.

Figure 531 shows a nucleotide sequence (SEQ ID NO:531) designated herein as DNA94477.

Figure 532 shows a nucleotide sequence (SEQ ID NO:532) designated herein as DNA94518.

Figure 533 shows a nucleotide sequence (SEQ ID NO:533) designated herein as DNA94533.

Figure 534 shows a nucleotide sequence (SEQ ID NO:534) designated herein as DNA95370.

Figure 535 shows a nucleotide sequence (SEQ ID NO:535) designated herein as DNA97358.

Figure 536 shows a nucleotide sequence (SEQ ID NO:536) designated herein as DNA97374.

Figure 537 shows a nucleotide sequence (SEQ ID NO:537) designated herein as DNA97470.

Figure 538 shows a nucleotide sequence (SEQ ID NO:538) designated herein as DNA97581.

Figure 539 shows a nucleotide sequence (SEQ ID NO:539) designated herein as DNA97767.

Figure 540 shows a nucleotide sequence (SEQ ID NO:540) designated herein as DNA97842.

Figure 541 shows a nucleotide sequence (SEQ ID NO:541) designated herein as DNA97949.

Figure 542 shows a nucleotide sequence (SEQ ID NO:542) designated herein as DNA97987.

Figure 543 shows a nucleotide sequence (SEQ ID NO:543) designated herein as DNA97995.

Figure 544 shows a nucleotide sequence (SEQ ID NO:544) designated herein as DNA98293.

Figure 545 shows a nucleotide sequence (SEQ ID NO:545) designated herein as DNA98294.

Figure 546 shows a nucleotide sequence (SEQ ID NO:546) designated herein as DNA98346.

Figure 547 shows a nucleotide sequence (SEQ ID NO:547) designated herein as DNA98360.

Figure 548 shows a nucleotide sequence (SEQ ID NO:548) designated herein as DNA98829.

Figure 549 shows a nucleotide sequence (SEQ ID NO:549) designated herein as DNA101514.

Figure 550 shows a nucleotide sequence (SEQ ID NO:550) designated herein as DNA101572.

Figure 551 shows a nucleotide sequence (SEQ ID NO:551) designated herein as DNA101580.

Figure 552 shows a nucleotide sequence (SEQ ID NO:552) designated herein as DNA101595.

Figure 553 shows a nucleotide sequence (SEQ ID NO:553) designated herein as DNA101633.

Figure 554 shows a nucleotide sequence (SEQ ID NO:554) designated herein as DNA101717.

Figure 555 shows a nucleotide sequence (SEQ ID NO:555) designated herein as DNA101768.

Figure 556 shows a nucleotide sequence (SEQ ID NO:556) designated herein as DNA107332.

Figure 557 shows a nucleotide sequence (SEQ ID NO:557) designated herein as DNA43499.

Figure 558 shows a nucleotide sequence (SEQ ID NO:558) designated herein as DNA45713.

Figure 559 shows a nucleotide sequence (SEQ ID NO:559) designated herein as DNA46089.

Figure 560 shows a nucleotide sequence (SEQ ID NO:560) designated herein as DNA68256.

Figure 561 shows a nucleotide sequence (SEQ ID NO:561) designated herein as DNA70305.

Figure 562 shows a nucleotide sequence (SEQ ID NO:562) designated herein as DNA82953.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

I. Definitions

The term "SRT polypeptide" when used herein encompasses "native sequence SRT polypeptides" and "SRT polypeptide variants" (which are further defined herein). "SRT" is a designation given to those polypeptides which are encoded by the nucleic acid molecules shown in the accompanying figures and variants thereof, nucleic acid molecules comprising the sequence shown in the accompanying figures and variants thereof as well as fragments of the above. The SRT polypeptides of the invention may be isolated from a variety of sources, such as from human tissue types or from another source, or prepared by recombinant and/or synthetic

methods.

A "native sequence" SRT polypeptide comprises a polypeptide having the same amino acid sequence as the corresponding SRT polypeptide derived from nature. Such native sequence SRT polypeptides can be isolated from nature or can be produced by recombinant and/or synthetic means. The term "native sequence SRT polypeptide" specifically encompasses naturally-occurring truncated or secreted forms (e.g., an extracellular domain sequence), naturally-occurring variant forms (e.g., alternatively spliced forms) and naturally-occurring allelic variants of the polypeptide.

An SRT polypeptide "extracellular domain" or "ECD" refers to a form of the SRT polypeptide which is essentially free of the transmembrane and cytoplasmic domains. Ordinarily, an SRT polypeptide ECD will have less than about 1% of such transmembrane and/or cytoplasmic domains and preferably, will have less than about 0.5% of such domains. It will be understood that any transmembrane domain(s) identified for the SRT polypeptides of the present invention are identified pursuant to criteria routinely employed in the art for identifying that type of hydrophobic domain. The exact boundaries of a transmembrane domain may vary but most likely by no more than about 5 amino acids at either end of the domain as initially identified.

"Variant SRT polypeptide" means an active SRT polypeptide as defined below having at least about 80% amino acid sequence identity with the amino acid sequence of a specifically derived fragment of any other polypeptide which will be specifically recited. Such variant SRT polypeptides include, for instance, SRT polypeptides wherein one or more amino acid residues are added, or deleted, at the N- and/or C-terminus, as well as within one or more internal domains, of the full-length amino acid sequence. Ordinarily, a variant SRT polypeptide will have at least about 80% amino acid sequence identity, more preferably at least about 81% amino acid sequence identity, more preferably at least about 82% amino acid sequence identity, more preferably at least about 83% amino acid sequence identity, more preferably at least about 84% amino acid sequence identity, more preferably at least about 85% amino acid sequence identity, more preferably at least about 86% amino acid sequence identity, more preferably at least about 87% amino acid sequence identity, more preferably at least about 88% amino acid sequence identity, more preferably at least about 89% amino acid sequence identity, more preferably at least about 90% amino acid sequence identity, more preferably at least about 91% amino acid sequence identity, more preferably at least about 92% amino acid sequence identity, more preferably at least about 93% amino acid sequence identity, more preferably at least about 94%

amino acid sequence identity, more preferably at least about 95% amino acid sequence identity, more preferably at least about 96% amino acid sequence identity, more preferably at least about 97% amino acid sequence identity, more preferably at least about 98% amino acid sequence identity and yet more preferably at least about 99% amino acid sequence identity with an SRT polypeptide encoded by a nucleic acid molecule shown in one of the accompanying figures or a specified fragment thereof. SRT variant polypeptides do not encompass the native SRT polypeptide sequence. Ordinarily, SRT variant polypeptides are at least about 10 amino acids in length, often at least about 20 amino acids in length, more often at least about 30 amino acids in length, more often at least about 40 amino acids in length, more often at least about 50 amino acids in length, more often at least about 60 amino acids in length, more often at least about 70 amino acids in length, more often at least about 80 amino acids in length, more often at least about 90 amino acids in length, more often at least about 100 amino acids in length, more often at least about 150 amino acids in length, more often at least about 200 amino acids in length, more often at least about 250 amino acids in length, more often at least about 300 amino acids in length, or more.

"Percent (%) amino acid sequence identity" with respect to the SRT polypeptide sequences identified herein is defined as the percentage of amino acid residues in a candidate sequence that are identical with the amino acid residues in a SRT sequence, after aligning the sequences and introducing gaps, if necessary, to achieve the maximum percent sequence identity, and not considering any conservative substitutions as part of the sequence identity. Alignment for purposes of determining percent amino acid sequence identity can be achieved in various ways that are within the skill in the art, for instance, using publicly available computer software such as BLAST, BLAST-2, ALIGN, ALIGN-2 or Megalign (DNASTAR) software. Those skilled in the art can determine appropriate parameters for measuring alignment, including any algorithms needed to achieve maximal alignment over the full-length of the sequences being compared. For purposes herein, however, % amino acid sequence identity values are obtained as described below by using the sequence comparison computer program ALIGN-2, wherein the complete source code for the ALIGN-2 program is provided in Table 1. The ALIGN-2 sequence comparison computer program was authored by Genentech, Inc. and the source code shown in Table 1 has been filed with user documentation in the U.S. Copyright Office, Washington D.C., 20559, where it is registered under U.S. Copyright Registration No. TXU510087. The ALIGN-2 program is publicly available through Genentech, Inc., South San Francisco, California or may

be compiled from the source code provided in Table 1. The ALIGN-2 program should be compiled for use on a UNIX operating system, preferably digital UNIX V4.0D. All sequence comparison parameters are set by the ALIGN-2 program and do not vary.

For purposes herein, the % amino acid sequence identity of a given amino acid sequence A to, with, or against a given amino acid sequence B (which can alternatively be phrased as a given amino acid sequence A that has or comprises a certain % amino acid sequence identity to, with, or against a given amino acid sequence B) is calculated as follows:

100 times the fraction X/Y

where X is the number of amino acid residues scored as identical matches by the sequence alignment program ALIGN-2 in that program's alignment of A and B, and where Y is the total number of amino acid residues in B. It will be appreciated that where the length of amino acid sequence A is not equal to the length of amino acid sequence B, the % amino acid sequence identity of A to B will not equal the % amino acid sequence identity of B to A. As examples of % amino acid sequence identity calculations, Tables 2 and 3 demonstrate how to calculate the % amino acid sequence identity of the amino acid sequence designated "Comparison Protein" to the amino acid sequence designated "PRO".

Unless specifically stated otherwise, all % amino acid sequence identity values used herein are obtained as described above using the ALIGN-2 sequence comparison computer program. However, % amino acid sequence identity may also be determined using the sequence comparison program NCBI-BLAST2 (Altschul et al., Nucleic Acids Res. 25:3389-3402 (1997)). The NCBI-BLAST2 sequence comparison program may be downloaded from http://www.ncbi.nlm.nih.gov. NCBI-BLAST2 uses several search parameters, wherein all of those search parameters are set to default values including, for example, unmask = yes, strand = all, expected occurrences = 10, minimum low complexity length = 15/5, multi-pass e-value = 0.01, constant for multi-pass = 25, dropoff for final gapped alignment = 25 and scoring matrix = BLOSUM62.

In situations where NCBI-BLAST2 is employed for amino acid sequence comparisons, the % amino acid sequence identity of a given amino acid sequence A to, with, or against a given amino acid sequence B (which can alternatively be phrased as a given amino acid sequence A that has or comprises a certain % amino acid sequence identity to, with, or against a given amino

acid sequence B) is calculated as follows:

100 times the fraction X/Y

where X is the number of amino acid residues scored as identical matches by the sequence alignment program NCBI-BLAST2 in that program's alignment of A and B, and where Y is the total number of amino acid residues in B. It will be appreciated that where the length of amino acid sequence A is not equal to the length of amino acid sequence B, the % amino acid sequence identity of A to B will not equal the % amino acid sequence identity of B to A.

"SRT variant polynucleotide" or "SRT variant nucleic acid sequence" means a nucleic acid molecule which has at least about 80% nucleic acid sequence identity with any of the nucleic acid sequences shown in the accompanying figures or a specified fragment thereof. Ordinarily, a SRT variant polynucleotide will have at least about 80% nucleic acid sequence identity, more preferably at least about 81% nucleic acid sequence identity, more preferably at least about 82% nucleic acid sequence identity, more preferably at least about 83% nucleic acid sequence identity, more preferably at least about 84% nucleic acid sequence identity, more preferably at least about 85% nucleic acid sequence identity, more preferably at least about 86% nucleic acid sequence identity, more preferably at least about 87% nucleic acid sequence identity, more preferably at least about 88% nucleic acid sequence identity, more preferably at least about 89% nucleic acid sequence identity, more preferably at least about 90% nucleic acid sequence identity, more preferably at least about 91% nucleic acid sequence identity, more preferably at least about 92% nucleic acid sequence identity, more preferably at least about 93% nucleic acid sequence identity, more preferably at least about 94% nucleic acid sequence identity, more preferably at least about 95% nucleic acid sequence identity, more preferably at least about 96% nucleic acid sequence identity, more preferably at least about 97% nucleic acid sequence identity, more preferably at least about 98% nucleic acid sequence identity and yet more preferably at least about 99% nucleic acid sequence identity with any of the nucleic acid sequences shown in the accompanying figures or a specified fragment thereof. SRT polynucleotide variants do not encompass the native SRT nucleotide sequence.

Ordinarily, SRT variant polynucleotides are at least about 10 nucleotides in length, often at least about 15 nucleotides in length, often at least about 20 nucleotides in length, often at least about 25 nucleotides in length, often at least about 30 nucleotides in length, often at least about

35 nucleotides in length, often at least about 40 nucleotides in length, often at least about 45 nucleotides in length, often at least about 50 nucleotides in length, often at least about 55 nucleotides in length, often at least about 60 nucleotides in length, often at least about 65 nucleotides in length, often at least about 65 nucleotides in length, often at least about 70 nucleotides in length, often at least about 75 nucleotides in length, often at least about 80 nucleotides in length, often at least about 85 nucleotides in length, often at least about 90 nucleotides in length, often at least about 95 nucleotides in length, often at least about 100 nucleotides in length, or more.

"Percent (%) nucleic acid sequence identity" with respect to the SRT polypeptide-encoding nucleic acid sequences identified herein is defined as the percentage of nucleotides in a candidate sequence that are identical with the nucleotides in a SRT polypeptide-encoding nucleic acid sequence, after aligning the sequences and introducing gaps, if necessary, to achieve the maximum percent sequence identity. Alignment for purposes of determining percent nucleic acid sequence identity can be achieved in various ways that are within the skill in the art, for instance, using publicly available computer software such as BLAST, BLAST-2, ALIGN, ALIGN-2 or Megalign (DNASTAR) software. Those skilled in the art can determine appropriate parameters for measuring alignment, including any algorithms needed to achieve maximal alignment over the full-length of the sequences being compared. For purposes herein, however, % nucleic acid sequence identity values are obtained as described below by using the sequence comparison computer program ALIGN-2, wherein the complete source code for the ALIGN-2 program is provided in Table 1. The ALIGN-2 sequence comparison computer program was authored by Genentech, Inc. and the source code shown in Table 1 has been filed with user documentation in the U.S. Copyright Office, Washington D.C., 20559, where it is registered under U.S. Copyright Registration No. TXU510087. The ALIGN-2 program is publicly available through Genentech, Inc., South San Francisco, California or may be compiled from the source code provided in Table 1. The ALIGN-2 program should be compiled for use on a UNIX operating system, preferably digital UNIX V4.0D. All sequence comparison parameters are set by the ALIGN-2 program and do not vary.

For purposes herein, the % nucleic acid sequence identity of a given nucleic acid sequence C to, with, or against a given nucleic acid sequence D (which can alternatively be phrased as a given nucleic acid sequence C that has or comprises a certain % nucleic acid sequence identity to, with, or against a given nucleic acid sequence D) is calculated as follows:

100 times the fraction W/Z

where W is the number of nucleotides scored as identical matches by the sequence alignment program ALIGN-2 in that program's alignment of C and D, and where Z is the total number of nucleotides in D. It will be appreciated that where the length of nucleic acid sequence C is not equal to the length of nucleic acid sequence D, the % nucleic acid sequence identity of C to D will not equal the % nucleic acid sequence identity of D to C. As examples of % nucleic acid sequence identity calculations, Tables 4 and 5 demonstrate how to calculate the % nucleic acid sequence identity of the nucleic acid sequence designated "Comparison DNA" to the nucleic acid sequence designated "PRO-DNA".

Unless specifically stated otherwise, all % nucleic acid sequence identity values used herein are obtained as described above using the ALIGN-2 sequence comparison computer program. However, % nucleic acid sequence identity may also be determined using the sequence comparison program NCBI-BLAST2 (Altschul et al., Nucleic Acids Res. 25:3389-3402 (1997)). The NCBI-BLAST2 sequence comparison program may be downloaded from http://www.ncbi.nlm.nih.gov. NCBI-BLAST2 uses several search parameters, wherein all of those search parameters are set to default values including, for example, unmask = yes, strand = all, expected occurrences = 10, minimum low complexity length = 15/5, multi-pass e-value = 0.01, constant for multi-pass = 25, dropoff for final gapped alignment = 25 and scoring matrix = BLOSUM62.

In situations where NCBI-BLAST2 is employed for sequence comparisons, the % nucleic acid sequence identity of a given nucleic acid sequence C to, with, or against a given nucleic acid sequence D (which can alternatively be phrased as a given nucleic acid sequence C that has or comprises a certain % nucleic acid sequence identity to, with, or against a given nucleic acid sequence D) is calculated as follows:

100 times the fraction W/Z

where W is the number of nucleotides scored as identical matches by the sequence alignment program NCBI-BLAST2 in that program's alignment of C and D, and where Z is the total number of nucleotides in D. It will be appreciated that where the length of nucleic acid sequence C is not equal to the length of nucleic acid sequence D, the % nucleic acid sequence identity of

C to D will not equal the % nucleic acid sequence identity of D to C.

In other embodiments, SRT variant polynucleotides are nucleic acid molecules that encode an active SRT polypeptide and which are capable of hybridizing, preferably under stringent hybridization conditions, to any of the nucleotide sequences shown in the accompanying figures or their complements. SRT variant polypeptides may be those that are encoded by a SRT variant polynucleotide.

The term "positives", in the context of the amino acid sequence identity comparisons performed as described above, includes amino acid residues in the sequences compared that are not only identical, but also those that have similar properties. Amino acid residues that score a positive value to an amino acid residue of interest are those that are either identical to the amino acid residue of interest or are a preferred substitution (as defined in Table 6 below) of the amino acid residue of interest.

For purposes herein, the % value of positives of a given amino acid sequence A to, with, or against a given amino acid sequence B (which can alternatively be phrased as a given amino acid sequence A that has or comprises a certain % positives to, with, or against a given amino acid sequence B) is calculated as follows:

100 times the fraction X/Y

where X is the number of amino acid residues scoring a positive value as defined above by the sequence alignment program ALIGN-2 in that program's alignment of A and B, and where Y is the total number of amino acid residues in B. It will be appreciated that where the length of amino acid sequence A is not equal to the length of amino acid sequence B, the % positives of A to B will not equal the % positives of B to A.

"Isolated," when used to describe the various polypeptides disclosed herein, means polypeptide that has been identified and separated and/or recovered from a component of its natural environment. Preferably, the isolated polypeptide is free of association with all components with which it is naturally associated. Contaminant components of its natural environment are materials that would typically interfere with diagnostic or therapeutic uses for the polypeptide, and may include enzymes, hormones, and other proteinaceous or non-proteinaceous solutes. In preferred embodiments, the polypeptide will be purified (1) to a degree sufficient to obtain at least 15 residues of N-terminal or internal amino acid sequence by use of

a spinning cup sequenator, or (2) to homogeneity by SDS-PAGE under non-reducing or reducing conditions using Coomassie blue or, preferably, silver stain. Isolated polypeptide includes polypeptide *in situ* within recombinant cells, since at least one component of the SRT natural environment will not be present. Ordinarily, however, isolated polypeptide will be prepared by at least one purification step.

An "isolated" nucleic acid molecule encoding a SRT polypeptide is a nucleic acid molecule that is identified and separated from at least one contaminant nucleic acid molecule with which it is ordinarily associated in the natural source of the SRT-encoding nucleic acid. Preferably, the isolated nucleic is free of association with all components with which it is naturally associated. An isolated SRT-encoding nucleic acid molecule is other than in the form or setting in which it is found in nature. Isolated nucleic acid molecules therefore are distinguished from the SRT-encoding nucleic acid molecule as it exists in natural cells. However, an isolated nucleic acid molecule encoding a SRT polypeptide includes SRT-encoding nucleic acid molecules contained in cells that ordinarily express SRT where, for example, the nucleic acid molecule is in a chromosomal location different from that of natural cells.

The term "control sequences" refers to DNA sequences necessary for the expression of an operably linked coding sequence in a particular host organism. The control sequences that are suitable for prokaryotes, for example, include a promoter, optionally an operator sequence, and a ribosome binding site. Eukaryotic cells are known to utilize promoters, polyadenylation signals, and enhancers.

Nucleic acid is "operably linked" when it is placed into a functional relationship with another nucleic acid sequence. For example, DNA for a presequence or secretory leader is operably linked to DNA for a polypeptide if it is expressed as a preprotein that participates in the secretion of the polypeptide; a promoter or enhancer is operably linked to a coding sequence if it affects the transcription of the sequence; or a ribosome binding site is operably linked to a coding sequence if it is positioned so as to facilitate translation. Generally, "operably linked" means that the DNA sequences being linked are contiguous, and, in the case of a secretory leader, contiguous and in reading phase. However, enhancers do not have to be contiguous. Linking is accomplished by ligation at convenient restriction sites. If such sites do not exist, the synthetic oligonucleotide adaptors or linkers are used in accordance with conventional practice.

The term "antibody" is used in the broadest sense and specifically covers, for example, single anti-SRT monoclonal antibodies (including agonist, antagonist, and neutralizing

antibodies), anti-SRT antibody compositions with polyepitopic specificity, single chain anti-SRT antibodies, and fragments of anti-SRT antibodies (see below). The term "monoclonal antibody" as used herein refers to an antibody obtained from a population of substantially homogeneous antibodies, *i.e.*, the individual antibodies comprising the population are identical except for possible naturally-occurring mutations that may be present in minor amounts.

"Stringency" of hybridization reactions is readily determinable by one of ordinary skill in the art, and generally is an empirical calculation dependent upon probe length, washing temperature, and salt concentration. In general, longer probes require higher temperatures for proper annealing, while shorter probes need lower temperatures. Hybridization generally depends on the ability of denatured DNA to reanneal when complementary strands are present in an environment below their melting temperature. The higher the degree of desired homology between the probe and hybridizable sequence, the higher the relative temperature which can be used. As a result, it follows that higher relative temperatures would tend to make the reaction conditions more stringent, while lower temperatures less so. For additional details and explanation of stringency of hybridization reactions, see Ausubel *et al.*, Current Protocols in Molecular Biology, Wiley Interscience Publishers, (1995).

"Stringent conditions" or "high stringency conditions", as defined herein, may be identified by those that: (1) employ low ionic strength and high temperature for washing, for example 0.015 M sodium chloride/0.0015 M sodium citrate/0.1% sodium dodecyl sulfate at 50°C; (2) employ during hybridization a denaturing agent, such as formamide, for example, 50% (v/v) formamide with 0.1% bovine serum albumin/0.1% Ficoll/0.1% polyvinylpyrrolidone/50mM sodium phosphate buffer at pH 6.5 with 750 mM sodium chloride, 75 mM sodium citrate at 42°C; or (3) employ 50% formamide, 5 x SSC (0.75 M NaCl, 0.075 M sodium citrate), 50 mM sodium phosphate (pH 6.8), 0.1% sodium pyrophosphate, 5 x Denhardt's solution, sonicated salmon sperm DNA (50 μg/ml), 0.1% SDS, and 10% dextran sulfate at 42°C, with washes at 42°C in 0.2 x SSC (sodium chloride/sodium citrate) and 50% formamide at 55°C, followed by a high-stringency wash consisting of 0.1 x SSC containing EDTA at 55°C.

"Moderately stringent conditions" may be identified as described by Sambrook *et al.*, Molecular Cloning: A Laboratory Manual, New York: Cold Spring Harbor Press, 1989, and include the use of washing solution and hybridization conditions (e.g., temperature, ionic strength and %SDS) less stringent that those described above. An example of moderately stringent conditions is overnight incubation at 37°C in a solution comprising: 20% formamide, 5 x SSC

(150 mM NaCl, 15 mM trisodium citrate), 50 mM sodium phosphate (pH 7.6), 5 x Denhardt's solution, 10% dextran sulfate, and 20 mg/ml denatured sheared salmon sperm DNA, followed by washing the filters in 1 x SSC at about 37-50°C. The skilled artisan will recognize how to adjust the temperature, ionic strength, etc. as necessary to accommodate factors such as probe length and the like.

The term "epitope tagged" when used herein refers to a chimeric polypeptide comprising a SRT polypeptide fused to a "tag polypeptide". The tag polypeptide has enough residues to provide an epitope against which an antibody can be made, yet is short enough such that it does not interfere with activity of the polypeptide to which it is fused. The tag polypeptide preferably also is fairly unique so that the antibody does not substantially cross-react with other epitopes. Suitable tag polypeptides generally have at least six amino acid residues and usually between about 8 and 50 amino acid residues (preferably, between about 10 and 20 amino acid residues).

As used herein, the term "immunoadhesin" designates antibody-like molecules which combine the binding specificity of a heterologous protein (an "adhesin") with the effector functions of immunoglobulin constant domains. Structurally, the immunoadhesins comprise a fusion of an amino acid sequence with the desired binding specificity which is other than the antigen recognition and binding site of an antibody (i.e., is "heterologous"), and an immunoglobulin constant domain sequence. The adhesin part of an immunoadhesin molecule typically is a contiguous amino acid sequence comprising at least the binding site of a receptor or a ligand. The immunoglobulin constant domain sequence in the immunoadhesin may be obtained from any immunoglobulin, such as IgG-1, IgG-2, IgG-3, or IgG-4 subtypes, IgA (including IgA-1 and IgA-2), IgE, IgD or IgM.

"Active" or "activity" for the purposes herein refers to form(s) of SRT which retain a biological and/or an immunological activity of native or naturally-occurring SRT, wherein "biological" activity refers to a biological function (either inhibitory or stimulatory) caused by a native or naturally-occurring SRT other than the ability to induce the production of an antibody against an antigenic epitope possessed by a native or naturally-occurring SRT and an "immunological" activity refers to the ability to induce the production of an antibody against an antigenic epitope possessed by a native or naturally-occurring SRT.

The term "antagonist" is used in the broadest sense, and includes any molecule that partially or fully blocks, inhibits, or neutralizes a biological activity of a native SRT polypeptide disclosed herein. In a similar manner, the term "agonist" is used in the broadest sense and

includes any molecule that mimics a biological activity of a native SRT polypeptide disclosed herein. Suitable agonist or antagonist molecules specifically include agonist or antagonist antibodies or antibody fragments, fragments or amino acid sequence variants of native SRT polypeptides, peptides, small organic molecules, etc. Methods for identifying agonists or antagonists of a SRT polypeptide may comprise contacting a SRT polypeptide with a candidate agonist or antagonist molecule and measuring a detectable change in one or more biological activities normally associated with the SRT polypeptide.

"Treatment" refers to both therapeutic treatment and prophylactic or preventative measures, wherein the object is to prevent or slow down (lessen) the targeted pathologic condition or disorder. Those in need of treatment include those already with the disorder as well as those prone to have the disorder or those in whom the disorder is to be prevented.

"Chronic" administration refers to administration of the agent(s) in a continuous mode as opposed to an acute mode, so as to maintain the initial therapeutic effect (activity) for an extended period of time. "Intermittent" administration is treatment that is not consecutively done without interruption, but rather is cyclic in nature.

"Mammal" for purposes of treatment refers to any animal classified as a mammal, including humans, domestic and farm animals, and zoo, sports, or pet animals, such as dogs, cats, cattle, horses, sheep, pigs, goats, rabbits, etc. Preferably, the mammal is human.

Administration "in combination with" one or more further therapeutic agents includes simultaneous (concurrent) and consecutive administration in any order.

"Carriers" as used herein include pharmaceutically acceptable carriers, excipients, or stabilizers which are nontoxic to the cell or mammal being exposed thereto at the dosages and concentrations employed. Often the physiologically acceptable carrier is an aqueous pH buffered solution. Examples of physiologically acceptable carriers include buffers such as phosphate, citrate, and other organic acids; antioxidants including ascorbic acid; low molecular weight (less than about 10 residues) polypeptide; proteins, such as serum albumin, gelatin, or immunoglobulins; hydrophilic polymers such as polyvinylpyrrolidone; amino acids such as glycine, glutamine, asparagine, arginine or lysine; monosaccharides, disaccharides, and other carbohydrates including glucose, mannose, or dextrins; chelating agents such as EDTA; sugar alcohols such as mannitol or sorbitol; salt-forming counterions such as sodium; and/or nonionic surfactants such as TWEEN™, polyethylene glycol (PEG), and PLURONICS™.

"Antibody fragments" comprise a portion of an intact antibody, preferably the antigen

binding or variable region of the intact antibody. Examples of antibody fragments include Fab, Fab', $F(ab')_2$, and Fv fragments; diabodies; linear antibodies (Zapata et al., Protein Eng. 8(10): 1057-1062 [1995]); single-chain antibody molecules; and multispecific antibodies formed from antibody fragments.

Papain digestion of antibodies produces two identical antigen-binding fragments, called "Fab" fragments, each with a single antigen-binding site, and a residual "Fc" fragment, a designation reflecting the ability to crystallize readily. Pepsin treatment yields an $F(ab')_2$ fragment that has two antigen-combining sites and is still capable of cross-linking antigen.

"Fv" is the minimum antibody fragment which contains a complete antigen-recognition and -binding site. This region consists of a dimer of one heavy- and one light-chain variable domain in tight, non-covalent association. It is in this configuration that the three CDRs of each variable domain interact to define an antigen-binding site on the surface of the VH-VL dimer. Collectively, the six CDRs confer antigen-binding specificity to the antibody. However, even a single variable domain (or half of an Fv comprising only three CDRs specific for an antigen) has the ability to recognize and bind antigen, although at a lower affinity than the entire binding site.

The Fab fragment also contains the constant domain of the light chain and the first constant domain (CH1) of the heavy chain. Fab fragments differ from Fab' fragments by the addition of a few residues at the carboxy terminus of the heavy chain CH1 domain including one or more cysteines from the antibody hinge region. Fab'-SH is the designation herein for Fab' in which the cysteine residue(s) of the constant domains bear a free thiol group. $F(ab')_2$ antibody fragments originally were produced as pairs of Fab' fragments which have hinge cysteines between them. Other chemical couplings of antibody fragments are also known.

The "light chains" of antibodies (immunoglobulins) from any vertebrate species can be assigned to one of two clearly distinct types, called kappa and lambda, based on the amino acid sequences of their constant domains.

Depending on the amino acid sequence of the constant domain of their heavy chains, immunoglobulins can be assigned to different classes. There are five major classes of immunoglobulins: IgA, IgD, IgE, IgG, and IgM, and several of these may be further divided into subclasses (isotypes), e.g., IgG1, IgG2, IgG3, IgG4, IgA, and IgA2.

"Single-chain Fv" or "sFv" antibody fragments comprise the VH and VL domains of antibody, wherein these domains are present in a single polypeptide chain. Preferably, the Fv

polypeptide further comprises a polypeptide linker between the VH and VL domains which enables the sFv to form the desired structure for antigen binding. For a review of sFv, see Pluckthun in The Pharmacology of Monoclonal Antibodies, vol. 113, Rosenburg and Moore eds., Springer-Verlag, New York, pp. 269-315 (1994).

The term "diabodies" refers to small antibody fragments with two antigen-binding sites, which fragments comprise a heavy-chain variable domain (VH) connected to a light-chain variable domain (VL) in the same polypeptide chain (VH - VL). By using a linker that is too short to allow pairing between the two domains on the same chain, the domains are forced to pair with the complementary domains of another chain and create two antigen-binding sites. Diabodies are described more fully in, for example, EP 404,097; WO 93/11161; and Hollinger et al., Proc. Natl. Acad. Sci. USA, 90:6444-6448 (1993).

An "isolated" antibody is one which has been identified and separated and/or recovered from a component of its natural environment. Contaminant components of its natural environment are materials which would interfere with diagnostic or therapeutic uses for the antibody, and may include enzymes, hormones, and other proteinaceous or nonproteinaceous solutes. In preferred embodiments, the antibody will be purified (1) to greater than 95% by weight of antibody as determined by the Lowry method, and most preferably more than 99% by weight, (2) to a degree sufficient to obtain at least 15 residues of N-terminal or internal amino acid sequence by use of a spinning cup sequenator, or (3) to homogeneity by SDS-PAGE under reducing or nonreducing conditions using Coomassie blue or, preferably, silver stain. Isolated antibody includes the antibody in situ within recombinant cells since at least one component of the antibody's natural environment will not be present. Ordinarily, however, isolated antibody will be prepared by at least one purification step.

An antibody that "specifically binds to" or is "specific for" a particular polypeptide or an epitope on a particular polypeptide is one that binds to that particular polypeptide or epitope on a particular polypeptide without substantially binding to any other polypeptide or polypeptide epitope.

The word "label" when used herein refers to a detectable compound or composition which is conjugated directly or indirectly to the antibody so as to generate a "labeled" antibody. The label may be detectable by itself (e.g. radioisotope labels or fluorescent labels) or, in the case of an enzymatic label, may catalyze chemical alteration of a substrate compound or composition which is detectable.

By "solid phase" is meant a non-aqueous matrix to which the antibody of the present invention can adhere. Examples of solid phases encompassed herein include those formed partially or entirely of glass (e.g., controlled pore glass), polysaccharides (e.g., agarose), polyacrylamides, polystyrene, polyvinyl alcohol and silicones. In certain embodiments, depending on the context, the solid phase can comprise the well of an assay plate; in others it is a purification column (e.g., an affinity chromatography column). This term also includes a discontinuous solid phase of discrete particles, such as those described in U.S. Patent No. 4,275,149.

A "liposome" is a small vesicle composed of various types of lipids, phospholipids and/or surfactant which is useful for delivery of a drug (such as a SRT polypeptide or antibody thereto) to a mammal. The components of the liposome are commonly arranged in a bilayer formation, similar to the lipid arrangement of biological membranes.

A "small molecule" is defined herein to have a molecular weight below about 500 Daltons.

An "oligonucleotide" or "oligomer" is a stretch of nucleotide residues which has a sufficient number of bases to be used in a polymerase chain reaction (PCR). These sequences are based on (or designed from) genomic or cDNA sequences and may be used to amplify, confirm, or reveal the presence of an identical, similar or complementary DNA or RNA in a particular cell or tissue. Oligonucleotides or oligomers comprise portions of a DNA sequence having at least about 10 nucleotides as described above. Oligonucleotides may be chemically synthesized and may be used as probes.

"Probes" are nucleic acid sequences of variable length, preferably between about 10 and as many as about 6000 nucleotides, depending upon use. They are used in the detection of identical, similar or complementary nucleic acid sequences. Longer length probes are usually obtained from a natural or recombinant source, are highly specific and are often much slower to hybridize to a target nucleic acid than are oligomers. Probes may be single- or double-stranded and may be carefully designaed to have specificity in PCR, hybridization membrane-based, or ELISA-like technologies.

"Detectably labeled" with regard to a nucleic acid molecule of the present invention means that the molecule has attached thereto, either covalently or non-covalently, a compound which is detectable such as, for example, radionuclides, enzymes, fluorescent, chemi-luminescent, or chromogenic agents. Detectable labels associate with, establish the

presence of, and may allow quantification of a particular nucleic or amino acid sequence.

A "portion" or "fragment" of a polynucleotide or nucleic acid molecule comprises all or any part of the nucleotide sequence having fewer nucleotides than about 6 kb, preferably fewer than about 1 kb which can be used as a probe. Such probes may be labelled with detectable labels using nick translation, Klenow fill-in reaction, PCR or other methods well known in the art. After pretesting to optimize reaction conditions and to eliminate false positives, nucleic acid probes may be used in Southern, Northern or in situ hybridizations to determine whether DNA or RNA encoding the protein is present in a biological sample, cell type, tissue, organ or organism.

Table 1

```
/*
 *
 * C-C increased from 12 to 15
 * Z is average of EQ
 * B is average of ND
 * match with stop is _M; stop-stop = 0; J (joker) match = 0
 */
#define  _M      -8      /* value of a match with a stop */

int      _day[26][26] = {
/*     A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z */
/* A */   { 2, 0,-2, 0, 0,-4, 1,-1,-1, 0,-1,-2,-1, 0,_M, 1, 0,-2, 1, 1, 0, 0,-6, 0,-3, 0},
/* B */   { 0, 3,-4, 3, 2,-5, 0, 1,-2, 0, 0,-3,-2, 2,_M,-1, 1, 0, 0, 0, 0,-2,-5, 0,-3, 1},
/* C */   {-2,-4,15,-5,-5,-4,-3,-3,-2, 0,-5,-6,-5,-4,_M,-3,-5,-4, 0,-2, 0,-2,-8, 0, 0,-5},
/* D */   { 0, 3,-5, 4, 3,-6, 1, 1,-2, 0, 0,-4,-3, 2,_M,-1, 2,-1, 0, 0, 0,-2,-7, 0,-4, 2},
/* E */   { 0, 2,-5, 3, 4,-5, 0, 1,-2, 0, 0,-3,-2, 1,_M,-1, 2,-1, 0, 0, 0,-2,-7, 0,-4, 3},
/* F */   {-4,-5,-4,-6,-5, 9,-5,-2, 1, 0,-5, 2, 0,-4,_M,-5,-5,-4,-3,-3, 0,-1, 0, 0, 7,-5},
/* G */   { 1, 0,-3, 1, 0,-5, 5,-2,-3, 0,-2,-4,-3, 0,_M,-1,-1,-3, 1, 0, 0,-1,-7, 0,-5, 0},
/* H */   {-1, 1,-3, 1, 1,-2,-2, 6,-2, 0, 0,-2,-2, 2,_M, 0, 3, 2,-1,-1, 0,-2,-3, 0, 0, 2},
/* I */   {-1,-2,-2,-2,-2, 1,-3,-2, 5, 0,-2, 2, 2,-2,_M,-2,-2,-2,-1, 0, 0, 4,-5, 0,-1,-2},
/* J */   { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* K */   {-1, 0,-5, 0, 0,-5,-2, 0,-2, 0, 5,-3, 0, 1,_M,-1, 1, 3, 0, 0, 0,-2,-3, 0,-4, 0},
/* L */   {-2,-3,-6,-4,-3, 2,-4,-2, 2, 0,-3, 6, 4,-3,_M,-3,-2,-3,-3,-1, 0, 2,-2, 0,-1,-2},
/* M */   {-1,-2,-5,-3,-2, 0,-3,-2, 2, 0, 0, 4, 6,-2,_M,-2,-1, 0,-2,-1, 0, 2,-4, 0,-2,-1},
/* N */   { 0, 2,-4, 2, 1,-4, 0, 2,-2, 0, 1,-3,-2, 2,_M,-1, 1, 0, 1, 0, 0,-2,-4, 0,-2, 1},
/* O */   {_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M, 0,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M},
/* P */   { 1,-1,-3,-1,-1,-5,-1, 0,-2, 0,-1,-3,-2,-1,_M, 6, 0, 0, 1, 0, 0,-1,-6, 0,-5, 0},
/* Q */   { 0, 1,-5, 2, 2,-5,-1, 3,-2, 0, 1,-2,-1, 1,_M, 0, 4, 1,-1,-1, 0,-2,-5, 0,-4, 3},
/* R */   {-2, 0,-4,-1,-1,-4,-3, 2,-2, 0, 3,-3, 0, 0,_M, 0, 1, 6, 0,-1, 0,-2, 2, 0,-4, 0},
/* S */   { 1, 0, 0, 0, 0,-3, 1,-1,-1, 0, 0,-3,-2, 1,_M, 1,-1, 0, 2, 1, 0,-1,-2, 0,-3, 0},
/* T */   { 1, 0,-2, 0, 0,-3, 0,-1, 0, 0, 0,-1,-1, 0,_M, 0,-1,-1, 1, 3, 0, 0,-5, 0,-3, 0},
/* U */   { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* V */   { 0,-2,-2,-2,-2,-1,-1,-2, 4, 0,-2, 2, 2,-2,_M,-1,-2,-2,-1, 0, 0, 4,-6, 0,-2,-2},
/* W */   {-6,-5,-8,-7,-7, 0,-7,-3,-5, 0,-3,-2,-4,-4,_M,-6,-5, 2,-2,-5, 0,-6,17, 0, 0,-6},
/* X */   { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* Y */   {-3,-3, 0,-4,-4, 7,-5, 0,-1, 0,-4,-1,-2,-2,_M,-5,-4,-4,-3,-3, 0,-2, 0, 0,10,-4},
/* Z */   { 0, 1,-5, 2, 3,-5, 0, 2,-2, 0, 0,-2,-1, 1,_M, 0, 3, 0, 0, 0, 0,-2,-6, 0,-4, 4}
};
```

Table 1 (cont')

```
/*
 */
#include <stdio.h>
#include <ctype.h>

#define  MAXJMP      16      /* max jumps in a diag */
#define  MAXGAP      24      /* don't continue to penalize gaps larger than this */
#define  JMPS        1024    /* max jmps in an path */
#define  MX          4       /* save if there's at least MX-1 bases since last jmp */

#define  DMAT        3       /* value of matching bases */
#define  DMIS        0       /* penalty for mismatched bases */
#define  DINS0       8       /* penalty for a gap */
#define  DINS1       1       /* penalty per base */
#define  PINS0       8       /* penalty for a gap */
#define  PINS1       4       /* penalty per residue */

struct jmp {
        short           n[MAXJMP];      /* size of jmp (neg for dely) */
        unsigned short  x[MAXJMP];      /* base no. of jmp in seq x */
};                                      /* limits seq to 2^16 -1 */

struct diag {
        int             score;          /* score at last jmp */
        long            offset;         /* offset of prev block */
        short           ijmp;           /* current jmp index */
        struct jmp      jp;             /* list of jmps */
};

struct path {
        int     spc;            /* number of leading spaces */
        short   n[JMPS];        /* size of jmp (gap) */
        int     x[JMPS];        /* loc of jmp (last elem before gap) */
};

char            *ofile;                 /* output file name */
char            *namex[2];              /* seq names: getseqs() */
char            *prog;                  /* prog name for err msgs */
char            *seqx[2];               /* seqs: getseqs() */
int             dmax;                   /* best diag: nw() */
int             dmax0;                  /* final diag */
int             dna;                    /* set if dna: main() */
int             endgaps;                /* set if penalizing end gaps */
int             gapx, gapy;             /* total gaps in seqs */
int             len0, len1;             /* seq lens */
int             ngapx, ngapy;           /* total size of gaps */
int             smax;                   /* max score: nw() */
int             *xbm;                   /* bitmap for matching */
long            offset;                 /* current offset in jmp file */
struct  diag    *dx;                    /* holds diagonals */
struct  path    pp[2];                  /* holds path for seqs */

char            *calloc(), *malloc(), *index(), *strcpy();
char            *getseq(), *g_calloc();
```

Table 1 (cont')

```
/* Needleman-Wunsch alignment program
 *
 * usage: progs file1 file2
 *   where file1 and file2 are two dna or two protein sequences.
 *   The sequences can be in upper- or lower-case an may contain ambiguity
 *   Any lines beginning with ';', '>' or '<' are ignored
 *   Max file length is 65535 (limited by unsigned short x in the jmp struct)
 *   A sequence with 1/3 or more of its elements ACGTU is assumed to be DNA
 *   Output is in the file "align.out"
 *
 * The program may create a tmp file in /tmp to hold info about traceback.
 * Original version developed under BSD 4.3 on a vax 8650
 */
#include "nw.h"
#include "day.h"

static    _dbval[26] = {
          1,14,2,13,0,0,4,11,0,0,12,0,3,15,0,0,0,5,6,8,8,7,9,0,10,0
};

static    _pbval[26] = {
          1, 2|(1<<('D'-'A'))|(1<<('N'-'A')), 4, 8, 16, 32, 64,
          128, 256, 0xFFFFFFFF, 1<<10, 1<<11, 1<<12, 1<<13, 1<<14,
          1<<15, 1<<16, 1<<17, 1<<18, 1<<19, 1<<20, 1<<21, 1<<22,
          1<<23, 1<<24, 1<<25|(1<<('E'-'A'))|(1<<('Q'-'A'))
};

main(ac, av)                                                    main
          int       ac;
          char      *av[];
{
          prog = av[0];
          if (ac != 3) {
                    fprintf(stderr,"usage: %s file1 file2\n", prog);
                    fprintf(stderr,"where file1 and file2 are two dna or two protein sequences.\n");
                    fprintf(stderr,"The sequences can be in upper- or lower-case\n");
                    fprintf(stderr,"Any lines beginning with ';' or '<' are ignored\n");
                    fprintf(stderr,"Output is in the file \"align.out\"\n");
                    exit(1);
          }
          namex[0] = av[1];
          namex[1] = av[2];
          seqx[0] = getseq(namex[0], &len0);
          seqx[1] = getseq(namex[1], &len1);
          xbm = (dna)? _dbval : _pbval;

          endgaps = 0;                              /* 1 to penalize endgaps */
          ofile = "align.out";              /* output file */

          nw();             /* fill in the matrix, get the possible jmps */
          readjmps();       /* get the actual jmps */
          print();          /* print stats, alignment */

          cleanup(0);       /* unlink any tmp files */
}
```

Table 1 (cont')

```
/* do the alignment, return best score: main()
 * dna: values in Fitch and Smith, PNAS, 80, 1382-1386, 1983
 * pro: PAM 250 values
 * When scores are equal, we prefer mismatches to any gap, prefer
 * a new gap to extending an ongoing gap, and prefer a gap in seqx
 * to a gap in seq y.
 */
nw()                                                                    nw
{
        char        *px, *py;       /* seqs and ptrs */
        int         *ndely, *dely;  /* keep track of dely */
        int         ndelx, delx;    /* keep track of delx */
        int         *tmp;           /* for swapping row0, row1 */
        int         mis;            /* score for each type */
        int         ins0, ins1;     /* insertion penalties */
        register    id;             /* diagonal index */
        register    ij;             /* jmp index */
        register    *col0, *col1;   /* score for curr, last row */
        register    xx, yy;         /* index into seqs */

        dx = (struct diag *)g_calloc("to get diags", len0+len1+1, sizeof(struct diag));

        ndely = (int *)g_calloc("to get ndely", len1+1, sizeof(int));
        dely = (int *)g_calloc("to get dely", len1+1, sizeof(int));
        col0 = (int *)g_calloc("to get col0", len1+1, sizeof(int));
        col1 = (int *)g_calloc("to get col1", len1+1, sizeof(int));
        ins0 = (dna)? DINS0 : PINS0;
        ins1 = (dna)? DINS1 : PINS1;

        smax = -10000;

        if (endgaps) {
                for (col0[0] = dely[0] = -ins0, yy = 1; yy <= len1; yy++) {
                        col0[yy] = dely[yy] = col0[yy-1] - ins1;
                        ndely[yy] = yy;
                }
                col0[0] = 0;            /* Waterman Bull Math Biol 84 */
        }
        else
                for (yy = 1; yy <= len1; yy++)
                        dely[yy] = -ins0;

        /* fill in match matrix
         */
        for (px = seqx[0], xx = 1; xx <= len0; px++, xx++) {
                /* initialize first entry in col
                 */
                if (endgaps) {
                        if (xx == 1)
                                col1[0] = delx = -(ins0+ins1);
                        else
                                col1[0] = delx = col0[0] - ins1;
                        ndelx = xx;
                }
                else {
                        col1[0] = 0;
                        delx = -ins0;
                        ndelx = 0;
                }
```

Table 1 (cont')

...nw

```
for (py = seqx[1], yy = 1; yy <= len1; py++, yy++) {
        mis = col0[yy-1];
        if (dna)
                mis += (xbm[*px-'A']&xbm[*py-'A'])? DMAT : DMIS;
        else
                mis += _day[*px-'A'][*py-'A'];

        /* update penalty for del in x seq;
         * favor new del over ongong del
         * ignore MAXGAP if weighting endgaps
         */
        if (endgaps || ndely[yy] < MAXGAP) {
                if (col0[yy] - ins0 >= dely[yy]) {
                        dely[yy] = col0[yy] - (ins0+ins1);
                        ndely[yy] = 1;
                } else {
                        dely[yy] -= ins1;
                        ndely[yy]++;
                }
        } else {
                if (col0[yy] - (ins0+ins1) >= dely[yy]) {
                        dely[yy] = col0[yy] - (ins0+ins1);
                        ndely[yy] = 1;
                } else
                        ndely[yy]++;
        }

        /* update penalty for del in y seq;
         * favor new del over ongong del
         */
        if (endgaps || ndelx < MAXGAP) {
                if (col1[yy-1] - ins0 >= delx) {
                        delx = col1[yy-1] - (ins0+ins1);
                        ndelx = 1;
                } else {
                        delx -= ins1;
                        ndelx++;
                }
        } else {
                if (col1[yy-1] - (ins0+ins1) >= delx) {
                        delx = col1[yy-1] - (ins0+ins1);
                        ndelx = 1;
                } else
                        ndelx++;
        }

        /* pick the maximum score; we're favoring
         * mis over any del and delx over dely
         */
```

Table 1 (cont')

...nw

```
                id = xx - yy + len1 - 1;
                if (mis >= delx && mis >= dely[yy])
                        col1[yy] = mis;
                else if (delx >= dely[yy]) {
                        col1[yy] = delx;
                        ij = dx[id].ijmp;
                        if (dx[id].jp.n[0] && (!dna || (ndelx >= MAXJMP
                        && xx > dx[id].jp.x[ij]+MX) || mis > dx[id].score+DINS0)) {
                                dx[id].ijmp++;
                                if (++ij >= MAXJMP) {
                                        writejmps(id);
                                        ij = dx[id].ijmp = 0;
                                        dx[id].offset = offset;
                                        offset += sizeof(struct jmp) + sizeof(offset);
                                }
                        }
                        dx[id].jp.n[ij] = ndelx;
                        dx[id].jp.x[ij] = xx;
                        dx[id].score = delx;
                }
                else {
                        col1[yy] = dely[yy];
                        ij = dx[id].ijmp;
        if (dx[id].jp.n[0] && (!dna || (ndely[yy] >= MAXJMP
                        && xx > dx[id].jp.x[ij]+MX) || mis > dx[id].score+DINS0)) {
                                dx[id].ijmp++;
                                if (++ij >= MAXJMP) {
                                        writejmps(id);
                                        ij = dx[id].ijmp = 0;
                                        dx[id].offset = offset;
                                        offset += sizeof(struct jmp) + sizeof(offset);
                                }
                        }
                        dx[id].jp.n[ij] = -ndely[yy];
                        dx[id].jp.x[ij] = xx;
                        dx[id].score = dely[yy];
                }
                if (xx == len0 && yy < len1) {
                        /* last col
                         */
                        if (endgaps)
                                col1[yy] -= ins0+ins1*(len1-yy);
                        if (col1[yy] > smax) {
                                smax = col1[yy];
                                dmax = id;
                        }
                }
        }
        if (endgaps && xx < len0)
                col1[yy-1] -= ins0+ins1*(len0-xx);
        if (col1[yy-1] > smax) {
                smax = col1[yy-1];
                dmax = id;
        }
        tmp = col0; col0 = col1; col1 = tmp;
}
(void) free((char *)ndely);
(void) free((char *)dely);
(void) free((char *)col0);
```

```
    (void) free((char *)col1);
}
```

Table 1 (cont')

```
/*
 *
 * print() -- only routine visible outside this module
 *
 * static:
 * getmat() -- trace back best path, count matches: print()
 * pr_align() -- print alignment of described in array p[]: print()
 * dumpblock() -- dump a block of lines with numbers, stars: pr_align()
 * nums() -- put out a number line: dumpblock()
 * putline() -- put out a line (name, [num], seq, [num]): dumpblock()
 * stars() - -put a line of stars: dumpblock()
 * stripname() -- strip any path and prefix from a seqname
 */

#include "nw.h"

#define SPC     3
#define P_LINE  256     /* maximum output line */
#define P_SPC   3       /* space between name or num and seq */

extern  _day[26][26];
int     olen;           /* set output line length */
FILE    *fx;            /* output file */

print()                                                    print
{
        int     lx, ly, firstgap, lastgap;      /* overlap */

        if ((fx = fopen(ofile, "w")) == 0) {
                fprintf(stderr,"%s: can't write %s\n", prog, ofile);
                cleanup(1);
        }
        fprintf(fx, "<first sequence: %s (length = %d)\n", namex[0], len0);
        fprintf(fx, "<second sequence: %s (length = %d)\n", namex[1], len1);
        olen = 60;
        lx = len0;
        ly = len1;
        firstgap = lastgap = 0;
        if (dmax < len1 - 1) {          /* leading gap in x */
                pp[0].spc = firstgap = len1 - dmax - 1;
                ly -= pp[0].spc;
        }
        else if (dmax > len1 - 1) {     /* leading gap in y */
                pp[1].spc = firstgap = dmax - (len1 - 1);
                lx -= pp[1].spc;
        }
        if (dmax0 < len0 - 1) {         /* trailing gap in x */
                lastgap = len0 - dmax0 -1;
                lx -= lastgap;
        }
        else if (dmax0 > len0 - 1) {    /* trailing gap in y */
                lastgap = dmax0 - (len0 - 1);
                ly -= lastgap;
        }
        getmat(lx, ly, firstgap, lastgap);
        pr_align();
}
```

Table 1 (cont')

getmat

```
/*
 * trace back the best path, count matches
 */
static
getmat(lx, ly, firstgap, lastgap)
        int     lx, ly;                 /* "core" (minus endgaps) */
        int     firstgap, lastgap;      /* leading trailing overlap */
{
        int             nm, i0, i1, siz0, siz1;
        char            outx[32];
        double          pct;
        register        n0, n1;
        register char   *p0, *p1;

        /* get total matches, score
         */
        i0 = i1 = siz0 = siz1 = 0;
        p0 = seqx[0] + pp[1].spc;
        p1 = seqx[1] + pp[0].spc;
        n0 = pp[1].spc + 1;
        n1 = pp[0].spc + 1;

        nm = 0;
        while ( *p0 && *p1 ) {
                if (siz0) {
                        p1++;
                        n1++;
                        siz0--;
                }
                else if (siz1) {
                        p0++;
                        n0++;
                        siz1--;
                }
                else {
                        if (xbm[*p0-'A']&xbm[*p1-'A'])
                                nm++;
                        if (n0++ == pp[0].x[i0])
                                siz0 = pp[0].n[i0++];
                        if (n1++ == pp[1].x[i1])
                                siz1 = pp[1].n[i1++];
                        p0++;
                        p1++;
                }
        }

        /* pct homology:
         * if penalizing endgaps, base is the shorter seq
         * else, knock off overhangs and take shorter core
         */
        if (endgaps)
                lx = (len0 < len1)? len0 : len1;
        else
                lx = (lx < ly)? lx : ly;
        pct = 100.*(double)nm/(double)lx;
        fprintf(fx, "\n");
        fprintf(fx, "<%d match%s in an overlap of %d: %.2f percent similarity\n",
                nm, (nm == 1)? "" : "es", lx, pct);
```

Table 1 (cont')

...getmat

```
        fprintf(fx, "<gaps in first sequence: %d", gapx);
        if (gapx) {
                (void) sprintf(outx, " (%d %s%s)",
                        ngapx, (dna)? "base":"residue", (ngapx == 1)? "":"s");
                fprintf(fx,"%s", outx);

        fprintf(fx, ", gaps in second sequence: %d", gapy);
        if (gapy) {
                (void) sprintf(outx, " (%d %s%s)",
                        ngapy, (dna)? "base":"residue", (ngapy == 1)? "":"s");
                fprintf(fx,"%s", outx);
        }
        if (dna)
                fprintf(fx,
                "\n<score: %d (match = %d, mismatch = %d, gap penalty = %d + %d per base)\n",
                smax, DMAT, DMIS, DINS0, DINS1);
        else
                fprintf(fx,
                "\n<score: %d (Dayhoff PAM 250 matrix, gap penalty = %d + %d per residue)\n",
                smax, PINS0, PINS1);
        if (endgaps)
                fprintf(fx,
                "<endgaps penalized. left endgap: %d %s%s, right endgap: %d %s%s\n",
                firstgap, (dna)? "base" : "residue", (firstgap == 1)? "" : "s",
                lastgap, (dna)? "base" : "residue", (lastgap == 1)? "" : "s");
        else
                fprintf(fx, "<endgaps not penalized\n");
}

static          nm;             /* matches in core -- for checking */
static          lmax;           /* lengths of stripped file names */
static          ij[2];          /* jmp index for a path */
static          nc[2];          /* number at start of current line */
static          ni[2];          /* current elem number -- for gapping */
static          siz[2];
static char     *ps[2];         /* ptr to current element */
static char     *po[2];         /* ptr to next output char slot */
static char     out[2][P_LINE]; /* output line */
static char     star[P_LINE];   /* set by stars() */

/*
 * print alignment of described in struct path pp[]
 */
static
pr_align()
```

pr_align

```
{
        int             nn;     /* char count */
        int             more;
        register        i;

        for (i =  0, lmax = 0; i < 2; i++) {
                nn = stripname(namex[i]);
                if (nn > lmax)
                        lmax = nn;

                nc[i] = 1;
                ni[i] = 1;
                siz[i] = ij[i] = 0;
```

```
ps[i] = seqx[i];
po[i] = out[i];                    }
```

Table 1 (cont')


```
for (nn = nm = 0, more = 1; more; ) {
        for (i = more = 0; i < 2; i++) {
                /*
                 * do we have more of this sequence?
                 */
                if (!*ps[i])
                        continue;

                more++;

                if (pp[i].spc) {        /* leading space */
                        *po[i]++ = ' ';
                        pp[i].spc--;
                }
                else if (siz[i]) {      /* in a gap */
                        *po[i]++ = '-';
                        siz[i]--;
                }
                else {                  /* we're putting a seq element
                                         */
                        *po[i] = *ps[i];
                        if (islower(*ps[i]))
                                *ps[i] = toupper(*ps[i]);
                        po[i]++;
                        ps[i]++;

                        /*
                         * are we at next gap for this seq?
                         */
                        if (ni[i] == pp[i].x[ij[i]]) {
                                /*
                                 * we need to merge all gaps
                                 * at this location
                                 */
                                siz[i] = pp[i].n[ij[i]++];
                                while (ni[i] == pp[i].x[ij[i]])
                                        siz[i] += pp[i].n[ij[i]++];
                        }
                        ni[i]++;
                }
        }
        if (++nn == olen || !more && nn) {
                dumpblock();
                for (i = 0; i < 2; i++)
                        po[i] = out[i];
                nn = 0;
        }
}
}

/*
 * dump a block of lines, including numbers, stars: pr_align()
 */
static
dumpblock()
{
        register  i;
```

```
for (i = 0; i < 2; i++)
        *po[i]-- = '\0';
```

Table 1 (cont')

...dumpblock

```
        (void) putc('\n', fx);
        for (i = 0; i < 2; i++) {
5               if (*out[i] && (*out[i] != ' ' || *(po[i]) != ' ')) {
                        if (i == 0)
                                nums(i);
                        if (i == 0 && *out[1])
                                stars();
10                      putline(i);
                        if (i == 0 && *out[1])
                                fprintf(fx, star);
                        if (i == 1)
                                nums(i);
15              }
        }
}

/*
20 * put out a number line: dumpblock()
 */
static
nums(ix)                                                                    nums
        int     ix;     /* index in out[] holding seq line */
25 {
        char            nline[P_LINE];
        register        i, j;
        register char   *pn, *px, *py;

30      for (pn = nline, i = 0; i < lmax+P_SPC; i++, pn++)
                *pn = ' ';
        for (i = nc[ix], py = out[ix]; *py; py++, pn++) {
                if (*py == ' ' || *py == '-')
                        *pn = ' ';
35              else {
                        if (i%10 == 0 || (i == 1 && nc[ix] != 1)) {
                                j = (i < 0)? -i : i;
                                for (px = pn; j; j /= 10, px--)
                                        *px = j%10 + '0';
40                              if (i < 0)
                                        *px = '-';
                        }
                        else
                                *pn = ' ';
45                      i++;
                }
        }
        *pn = '\0';
        nc[ix] = i;
50      for (pn = nline; *pn; pn++)
                (void) putc(*pn, fx);
        (void) putc('\n', fx);
}

55 /*
 * put out a line (name, [num], seq, [num]): dumpblock()
 */
static
putline(ix)                                                                 putline
```

```
int     ix;                                     {
```

Table 1 (cont')

...putline

```
        int             i;
        register char   *px;

        for (px = namex[ix], i = 0; *px && *px != ':'; px++, i++)
                (void) putc(*px, fx);
        for (; i < lmax+P_SPC; i++)
                (void) putc(' ', fx);

        /* these count from 1:
         * ni[] is current element (from 1)
         * nc[] is number at start of current line
         */
        for (px = out[ix]; *px; px++)
                (void) putc(*px&0x7F, fx);
        (void) putc('\n', fx);
}

/*
 * put a line of stars (seqs always in out[0], out[1]): dumpblock()
 */
static
```

stars

```
stars()
{
        int             i;
        register char   *p0, *p1, cx, *px;

        if (!*out[0] || (*out[0] == ' ' && *(po[0]) == ' ') ||
            !*out[1] || (*out[1] == ' ' && *(po[1]) == ' '))
                return;
        px = star;
        for (i = lmax+P_SPC; i; i--)
                *px++ = ' ';

        for (p0 = out[0], p1 = out[1]; *p0 && *p1; p0++, p1++) {
                if (isalpha(*p0) && isalpha(*p1)) {

                        if (xbm[*p0-'A']&xbm[*p1-'A'])
                                cx = '*';
                                nm++;
                        }
                        else if (!dna && _day[*p0-'A'][*p1-'A'] > 0)
                                cx = '.';
                        else
                                cx = ' ';
                }
                else
                        cx = ' ';
                *px++ = cx;
        }
        *px++ = '\n';
        *px = '\0';
}
```

Table 1 (cont')

```
/*
 * strip path or prefix from pn, return len: pr_align()
 */
static
stripname(pn)                                                    stripname
        char    *pn;    /* file name (may be path) */
{
        register char   *px, *py;

        py = 0;
        for (px = pn; *px; px++)
                if (*px == '/')
                        py = px + 1;
        if (py)
                (void) strcpy(pn, py);
        return(strlen(pn));

}
```

79

Table 1 (cont')

```
/*
 * cleanup() -- cleanup any tmp file
 * getseq() -- read in seq, set dna, len, maxlen
 * g_calloc() -- calloc() with error checkin
 * readjmps() -- get the good jmps, from tmp file if necessary
 * writejmps() -- write a filled array of jmps to a tmp file: nw()
 */
#include "nw.h"
#include <sys/file.h>

char    *jname = "/tmp/homgXXXXXX";             /* tmp file for jmps */
FILE    *fj;

int     cleanup();                              /* cleanup tmp file */
long    lseek();

/*
 * remove any tmp file if we blow
 */
cleanup(i)                                                          cleanup
        int     i;
{
        if (fj)
                (void) unlink(jname);
        exit(i);
}

/*
 * read, return ptr to seq, set dna, len, maxlen
 * skip lines starting with ';', '<', or '>'
 * seq in upper or lower case
 */
char    *
getseq(file, len)                                                   getseq
        char    *file;  /* file name */
        int     *len;   /* seq len */
{
        char            line[1024], *pseq;
        register char   *px, *py;
        int             natgc, tlen;
        FILE            *fp;

        if ((fp = fopen(file,"r")) == 0) {
                fprintf(stderr,"%s: can't read %s\n", prog, file);
                exit(1);
        }
        tlen = natgc = 0;
        while (fgets(line, 1024, fp)) {
                if (*line == ';' || *line == '<' || *line == '>')
                        continue;
                for (px = line; *px != '\n'; px++)
                        if (isupper(*px) || islower(*px))
                                tlen++;
        }
        if ((pseq = malloc((unsigned)(tlen+6))) == 0) {
                fprintf(stderr,"%s: malloc() failed to get %d bytes for %s\n", prog, tlen+6, file);
                exit(1);
        }
        pseq[0] = pseq[1] = pseq[2] = pseq[3] = '\0';
```

Table 1 (cont')

...getseq

```
        py = pseq + 4;
        *len = tlen;
        rewind(fp);

        while (fgets(line, 1024, fp)) {
                if (*line == ';' || *line == '<' || *line == '>')
                        continue;
                for (px = line; *px != '\n'; px++){
                        if (isupper(*px))
                                *py++ = *px;
                        else if (islower(*px))
                                *py++ = toupper(*px);
                        if (index("ATGCU",*(py-1)))
                                natgc++;
                }
        }
        *py++ = '\0';
        *py = '\0';
        (void) fclose(fp);
        dna = natgc > (tlen/3);
        return(pseq+4);
}
```

g_calloc

```
char    *
g_calloc(msg, nx, sz)
        char    *msg;           /* program, calling routine */
        int     nx, sz;         /* number and size of elements */
{
        char            *px, *calloc();

        if ((px = calloc((unsigned)nx, (unsigned)sz)) == 0) {
                if (*msg) {
                        fprintf(stderr, "%s: g_calloc() failed %s (n=%d, sz=%d)\n", prog, msg, nx, sz);
                        exit(1);
                }
        }
        return(px);
}

/*
 * get final jmps from dx[] or tmp file, set pp[], reset dmax: main()
 */
```

readjmps

```
readjmps()
{
        int             fd = -1;
        int             siz, i0, i1;
        register        i, j, xx;

        if (fj) {
                (void) fclose(fj);
                if ((fd = open(jname, O_RDONLY, 0)) < 0) {
                        fprintf(stderr, "%s: can't open() %s\n", prog, jname);
                        cleanup(1);
                }
        }
        for (i = i0 = i1 = 0, dmax0 = dmax, xx = len0; ; i++) {
                while (1) {
                        for (j = dx[dmax].ijmp; j >= 0 && dx[dmax].jp.x[j] >= xx; j--)
```

Table 1 (cont')

...readjmps

```
                        if (j < 0 && dx[dmax].offset && fj) {
                                (void) lseek(fd, dx[dmax].offset, 0);
                                (void) read(fd, (char *)&dx[dmax].jp, sizeof(struct jmp));
                                (void) read(fd, (char *)&dx[dmax].offset, sizeof(dx[dmax].offset));
                                dx[dmax].ijmp = MAXJMP-1;
                        }
                        else
                                break;
                }
                if (i >= JMPS) {
                        fprintf(stderr, "%s: too many gaps in alignment\n", prog);
                        cleanup(1);
                }
                if (j >= 0) {
                        siz = dx[dmax].jp.n[j];
                        xx = dx[dmax].jp.x[j];
                        dmax += siz;
                        if (siz < 0) {                  /* gap in second seq */
                                pp[1].n[i1] = -siz;
                                xx += siz;
                                /* id = xx - yy + len1 - 1
                                 */
                                pp[1].x[i1] = xx - dmax + len1 - 1;
                                gapy++;
                                ngapy -= siz;
/* ignore MAXGAP when doing endgaps */
                                siz = (-siz < MAXGAP || endgaps)? -siz : MAXGAP;
                                i1++;
                        }
                        else if (siz > 0) {     /* gap in first seq */
                                pp[0].n[i0] = siz;
                                pp[0].x[i0] = xx;
                                gapx++;
                                ngapx += siz;
/* ignore MAXGAP when doing endgaps */
                                siz = (siz < MAXGAP || endgaps)? siz : MAXGAP;
                                i0++;
                        }
                }
                else
                        break;
        }

        /* reverse the order of jmps
         */
        for (j = 0, i0--; j < i0; j++, i0--) {
                i = pp[0].n[j]; pp[0].n[j] = pp[0].n[i0]; pp[0].n[i0] = i;
                i = pp[0].x[j]; pp[0].x[j] = pp[0].x[i0]; pp[0].x[i0] = i;
        }
        for (j = 0, i1--; j < i1; j++, i1--) {
                i = pp[1].n[j]; pp[1].n[j] = pp[1].n[i1]; pp[1].n[i1] = i;
                i = pp[1].x[j]; pp[1].x[j] = pp[1].x[i1]; pp[1].x[i1] = i;
        }
        if (fd >= 0)
                (void) close(fd);
        if (fj) {
                (void) unlink(jname);
                fj = 0;
                offset = 0;
```

Table 1 (cont')

```
/*
 * write a filled jmp struct offset of the prev one (if any): nw()
 */
writejmps(ix)                                                          writejmps
        int     ix;
{
        char    *mktemp();

        if (!fj) {
                if (mktemp(jname) < 0) {
                        fprintf(stderr, "%s: can't mktemp() %s\n", prog, jname);
                        cleanup(1);
                }
                if ((fj = fopen(jname, "w")) == 0) {
                        fprintf(stderr, "%s: can't write %s\n", prog, jname);
                        exit(1);
                }
        }
        (void) fwrite((char *)&dx[ix].jp, sizeof(struct jmp), 1, fj);
        (void) fwrite((char *)&dx[ix].offset, sizeof(dx[ix].offset), 1, fj);
}
```

Table 2

PRO	XXXXXXXXXXXXXXX	(Length = 15 amino acids)
Comparison Protein	XXXXXYYYYYYY	(Length = 12 amino acids)

% amino acid sequence identity =

(the number of identically matching amino acid residues between the two polypeptide sequences as determined by ALIGN-2) divided by (the total number of amino acid residues of the PRO polypeptide) =

5 divided by 15 = 33.3%

Table 3

PRO	XXXXXXXXXX	(Length = 10 amino acids)
Comparison Protein	XXXXXYYYYYYZZYZ	(Length = 15 amino acids)

% amino acid sequence identity =

(the number of identically matching amino acid residues between the two polypeptide sequences as determined by ALIGN-2) divided by (the total number of amino acid residues of the PRO polypeptide) =

5 divided by 10 = 50%

Table 4

PRO-DNA	NNNNNNNNNNNNNN	(Length = 14 nucleotides)
Comparison DNA	NNNNNNLLLLLLLLL	(Length = 16 nucleotides)

% nucleic acid sequence identity =

(the number of identically matching nucleotides between the two nucleic acid sequences as determined by ALIGN-2) divided by (the total number of nucleotides of the PRO-DNA nucleic acid sequence) =

6 divided by 14 = 42.9%

Table 5

PRO-DNA	NNNNNNNNNNNN	(Length = 12 nucleotides)
Comparison DNA	NNNNLLLVV	(Length = 9 nucleotides)

% nucleic acid sequence identity =

(the number of identically matching nucleotides between the two nucleic acid sequences as determined by ALIGN-2) divided by (the total number of nucleotides of the PRO-DNA nucleic acid sequence) =

4 divided by 12 = 33.3%

II. Compositions and Methods of the Invention

A. Full-length SRT Polypeptides

The present invention provides newly identified and isolated polynucleotide sequences encoding at least a portion of full-length human polypeptides referred to in the present application as SRT polypeptides. In particular, cDNAs encoding at least a portion of SRT polypeptides have been identified and isolated, as disclosed in further detail in the Examples below. For sake of simplicity, in the present specification the polypeptides encoded by nucleic acid molecules disclosed herein as well as all further native homologues and variants included in the foregoing definition of SRT, will be referred to as "SRT", regardless of their origin or mode of preparation.

B. SRT Polypeptide Variants

In addition to the native sequence SRT polypeptides described herein, it is contemplated that SRT variants can be prepared. SRT variants can be prepared by introducing appropriate nucleotide changes into the SRT DNA, and/or by synthesis of the desired SRT polypeptide. Those skilled in the art will appreciate that amino acid changes may alter post-translational processes of the SRT, such as changing the number or position of glycosylation sites or altering the membrane anchoring characteristics.

Variations in the native sequence SRT or in various domains of the SRT described herein, can be made, for example, using any of the techniques and guidelines for conservative and non-conservative mutations set forth, for instance, in U.S. Patent No. 5,364,934. Variations may be a substitution, deletion or insertion of one or more codons encoding the SRT that results in a change in the amino acid sequence of the SRT as compared with the native sequence SRT. Optionally the variation is by substitution of at least one amino acid with any other amino acid in one or more of the domains of the SRT. Guidance in determining which amino acid residue may be inserted, substituted or deleted without adversely affecting the desired activity may be found by comparing the sequence of the SRT with that of homologous known protein molecules and minimizing the number of amino acid sequence changes made in regions of high homology. Amino acid substitutions can be the result of replacing one amino acid with another amino acid having similar structural and/or chemical properties, such as the replacement of a leucine with a serine, i.e., conservative amino acid replacements. Insertions or deletions may optionally be in the range of about 1 to 5 amino acids. The variation allowed may be determined by

systematically making insertions, deletions or substitutions of amino acids in the sequence and testing the resulting variants for activity exhibited by the full-length or mature native sequence.

SRT polypeptide fragments are provided herein. Such fragments may be truncated at the N-terminus or C-terminus, or may lack internal residues, for example, when compared with a full-length native protein. Certain fragments lack amino acid residues that are not essential for a desired biological activity of the SRT polypeptide.

SRT fragments may be prepared by any of a number of conventional techniques. Desired peptide fragments may be chemically synthesized. An alternative approach involves generating SRT fragments by enzymatic digestion, e.g., by treating the protein with an enzyme known to cleave proteins at sites defined by particular amino acid residues, or by digesting the DNA with suitable restriction enzymes and isolating the desired fragment. Yet another suitable technique involves isolating and amplifying a DNA fragment encoding a desired polypeptide fragment, by polymerase chain reaction (PCR). Oligonucleotides that define the desired termini of the DNA fragment are employed at the 5' and 3' primers in the PCR. Preferably, SRT polypeptide fragments share at least one biological and/or immunological activity with the corresponding native SRT polypeptide.

In particular embodiments, conservative substitutions of interest are shown in Table 6 under the heading of preferred substitutions. If such substitutions result in a change in biological activity, then more substantial changes, denominated exemplary substitutions in Table 6, or as further described below in reference to amino acid classes, are introduced and the products screened.

Table 6

Original Residue	Exemplary Substitutions	Preferred Substitutions
Ala (A)	val; leu; ile	val
Arg (R)	lys; gln; asn	lys
Asn (N)	gln; his; lys; arg	gln
Asp (D)	glu	glu
Cys (C)	ser	ser
Gln (Q)	asn	asn
Glu (E)	asp	asp
Gly (G)	pro; ala	ala
His (H)	asn; gln; lys; arg	arg
Ile (I)	leu; val; met; ala; phe; norleucine	leu

	Leu (L)	norleucine; ile; val; met; ala; phe	ile		
	Lys (K)	arg; gln; asn		arg	
	Met (M)	leu; phe; ile	leu		
	Phe (F)	leu; val; ile; ala; tyr		leu	
	Pro (P)	ala		ala	
	Ser (S)	thr		thr	
	Thr (T)	ser		ser	
	Trp (W)	tyr; phe		tyr	
	Tyr (Y)	trp; phe; thr; ser			phe
	Val (V)	ile; leu; met; phe; ala; norleucine		leu	

Substantial modifications in function or immunological identity of the SRT polypeptide are accomplished by selecting substitutions that differ significantly in their effect on maintaining (a) the structure of the polypeptide backbone in the area of the substitution, for example, as a sheet or helical conformation, (b) the charge or hydrophobicity of the molecule at the target site, or (c) the bulk of the side chain. Naturally occurring residues are divided into groups based on common side-chain properties:

(1) hydrophobic: norleucine, met, ala, val, leu, ile;

(2) neutral hydrophilic: cys, ser, thr;

(3) acidic: asp, glu;

(4) basic: asn, gln, his, lys, arg;

(5) residues that influence chain orientation: gly, pro; and

(6) aromatic: trp, tyr, phe.

Non-conservative substitutions will entail exchanging a member of one of these classes for another class. Such substituted residues also may be introduced into the conservative substitution sites or, more preferably, into the remaining (non-conserved) sites.

The variations can be made using methods known in the art such as oligonucleotide-mediated (site-directed) mutagenesis, alanine scanning, and PCR mutagenesis. Site-directed mutagenesis [Carter et al., Nucl. Acids Res., 13:4331 (1986); Zoller et al., Nucl. Acids Res., 10:6487 (1987)], cassette mutagenesis [Wells et al., Gene, 34:315 (1985)], restriction selection mutagenesis [Wells et al., Philos. Trans. R. Soc. London SerA, 317:415 (1986)] or other known techniques can be performed on the cloned DNA to produce the SRT variant DNA.

Scanning amino acid analysis can also be employed to identify one or more amino acids along a contiguous sequence. Among the preferred scanning amino acids are relatively small,

neutral amino acids. Such amino acids include alanine, glycine, serine, and cysteine. Alanine is typically a preferred scanning amino acid among this group because it eliminates the side-chain beyond the beta-carbon and is less likely to alter the main-chain conformation of the variant [Cunningham and Wells, Science, 244: 1081-1085 (1989)]. Alanine is also typically preferred because it is the most common amino acid. Further, it is frequently found in both buried and exposed positions [Creighton, The Proteins, (W.H. Freeman & Co., N.Y.); Chothia, J. Mol. Biol., 150:1 (1976)]. If alanine substitution does not yield adequate amounts of variant, an isoteric amino acid can be used.

C. Modifications of SRT Polypeptides

Covalent modifications of SRT polypeptides are included within the scope of this invention. One type of covalent modification includes reacting targeted amino acid residues of a SRT polypeptide with an organic derivatizing agent that is capable of reacting with selected side chains or the N- or C- terminal residues of the SRT. Derivatization with bifunctional agents is useful, for instance, for crosslinking SRT to a water-insoluble support matrix or surface for use in the method for purifying anti-SRT antibodies, and vice-versa. Commonly used crosslinking agents include, e.g., 1,1-bis(diazoacetyl)-2-phenylethane, glutaraldehyde, N-hydroxysuccinimide esters, for example, esters with 4-azidosalicylic acid, homobifunctional imidoesters, including disuccinimidyl esters such as 3,3'-dithiobis(succinimidylpropionate), bifunctional maleimides such as bis-N-maleimido-1,8-octane and agents such as methyl-3-[(p-azidophenyl)dithio]propioimidate.

Other modifications include deamidation of glutaminyl and asparaginyl residues to the corresponding glutamyl and aspartyl residues, respectively, hydroxylation of proline and lysine, phosphorylation of hydroxyl groups of seryl or threonyl residues, methylation of the α-amino groups of lysine, arginine, and histidine side chains [T.E. Creighton, Proteins: Structure and Molecular Properties, W.H. Freeman & Co., San Francisco, pp. 79-86 (1983)], acetylation of the N-terminal amine, and amidation of any C-terminal carboxyl group.

Another type of covalent modification of the SRT polypeptide included within the scope of this invention comprises altering the native glycosylation pattern of the polypeptide. "Altering the native glycosylation pattern" is intended for purposes herein to mean deleting one or more carbohydrate moieties found in native sequence SRT (either by removing the underlying glycosylation site or by deleting the glycosylation by chemical and/or enzymatic means), and/or

adding one or more glycosylation sites that are not present in the native sequence SRT. In addition, the phrase includes qualitative changes in the glycosylation of the native proteins, involving a change in the nature and proportions of the various carbohydrate moieties present.

Addition of glycosylation sites to the SRT polypeptide may be accomplished by altering the amino acid sequence. The alteration may be made, for example, by the addition of, or substitution by, one or more serine or threonine residues to the native sequence SRT (for O-linked glycosylation sites). The SRT amino acid sequence may optionally be altered through changes at the DNA level, particularly by mutating the DNA encoding the SRT polypeptide at preselected bases such that codons are generated that will translate into the desired amino acids.

Another means of increasing the number of carbohydrate moieties on the SRT polypeptide is by chemical or enzymatic coupling of glycosides to the polypeptide. Such methods are described in the art, e.g., in WO 87/05330 published 11 September 1987, and in Aplin and Wriston, CRC Crit. Rev. Biochem., pp. 259-306 (1981).

Removal of carbohydrate moieties present on the SRT polypeptide may be accomplished chemically or enzymatically or by mutational substitution of codons encoding for amino acid residues that serve as targets for glycosylation. Chemical deglycosylation techniques are known in the art and described, for instance, by Hakimuddin et al., Arch. Biochem. Biophys., 259:52 (1987) and by Edge et al., Anal. Biochem. 118:131 (1981). Enzymatic cleavage of carbohydrate moieties on polypeptides can be achieved by the use of a variety of endo- and exo-glycosidases as described by Thotakura et al., Meth. Enzymol., 138:350 (1987).

Another type of covalent modification of SRT comprises linking the SRT polypeptide to one of a variety of nonproteinaceous polymers, e.g., polyethylene glycol (PEG), polypropylene glycol, or polyoxyalkylenes, in the manner set forth in U.S. Patent Nos. 4,640,835; 4,496,689; 4,301,144; 4,670,417; 4,791,192 or 4,179,337.

The SRT polypeptides of the present invention may also be modified in a way to form chimeric molecules comprising SRT fused to another, heterologous polypeptide or amino acid sequence.

In one embodiment, such a chimeric molecule comprises a fusion of the SRT with a tag polypeptide which provides an epitope to which an anti-tag antibody can selectively bind. The epitope tag is generally placed at the amino- or carboxyl- terminus of the SRT. The presence of such epitope-tagged forms of the SRT can be detected using an antibody against the tag polypeptide. Also, provision of the epitope tag enables the SRT to be readily purified by affinity

purification using an anti-tag antibody or another type of affinity matrix that binds to the epitope tag. Various tag polypeptides and their respective antibodies are well known in the art. Examples include poly-histidine (poly-his) or poly-histidine-glycine (poly-his-gly) tags; the flu HA tag polypeptide and its antibody 12CA5 [Field et al., Mol. Cell. Biol., 8:2159-2165 (1988)]; the c-myc tag and the 8F9, 3C7, 6E10, G4, B7 and 9E10 antibodies thereto [Evan et al., Molecular and Cellular Biology, 5:3610-3616 (1985)]; and the Herpes Simplex virus glycoprotein D (gD) tag and its antibody [Paborsky et al., Protein Engineering, 3(6):547-553 (1990)]. Other tag polypeptides include the Flag-peptide [Hopp et al., BioTechnology, 6:1204-1210 (1988)]; the KT3 epitope peptide [Martin et al., Science, 255:192-194 (1992)]; an α-tubulin epitope peptide [Skinner et al., J. Biol. Chem., 266:15163-15166 (1991)]; and the T7 gene 10 protein peptide tag [Lutz-Freyermuth et al., Proc. Natl. Acad. Sci. USA, 87:6393-6397 (1990)].

In an alternative embodiment, the chimeric molecule may comprise a fusion of the SRT with an immunoglobulin or a particular region of an immunoglobulin. For a bivalent form of the chimeric molecule (also referred to as an "immunoadhesin"), such a fusion could be to the Fc region of an IgG molecule. The Ig fusions preferably include the substitution of a soluble (transmembrane domain deleted or inactivated) form of a SRT polypeptide in place of at least one variable region within an Ig molecule. In a particularly preferred embodiment, the immunoglobulin fusion includes the hinge, CH2 and CH3, or the hinge, CH1, CH2 and CH3 regions of an IgG1 molecule. For the production of immunoglobulin fusions see also US Patent No. 5,428,130 issued June 27, 1995.

D. Preparation of SRT Polypeptides

The description below relates primarily to production of SRT by culturing cells transformed or transfected with a vector containing SRT nucleic acid. It is, of course, contemplated that alternative methods, which are well known in the art, may be employed to prepare SRT. For instance, the SRT sequence, or portions thereof, may be produced by direct peptide synthesis using solid-phase techniques [see, e.g., Stewart et al., Solid-Phase Peptide Synthesis, W.H. Freeman Co., San Francisco, CA (1969); Merrifield, J. Am. Chem. Soc., 85:2149-2154 (1963)]. *In vitro* protein synthesis may be performed using manual techniques or by automation. Automated synthesis may be accomplished, for instance, using an Applied Biosystems Peptide Synthesizer (Foster City, CA) using manufacturer's instructions. Various portions of the SRT may be chemically synthesized separately and combined using chemical or

enzymatic methods to produce the full-length SRT.

1. Isolation of DNA Encoding SRT

DNA encoding SRT may be obtained from a cDNA library prepared from tissue believed to possess the SRT mRNA and to express it at a detectable level. Accordingly, human SRT DNA can be conveniently obtained from a cDNA library prepared from human tissue, such as described in the Examples. The SRT-encoding gene may also be obtained from a genomic library or by known synthetic procedures (e.g., automated nucleic acid synthesis).

Libraries can be screened with probes (such as antibodies to the SRT or oligonucleotides of at least about 20-80 bases) designed to identify the gene of interest or the protein encoded by it, wherein those probes may be based upon the polynucleotide sequences shown in the accompanying figures. Screening the cDNA or genomic library with the selected probe may be conducted using standard procedures, such as described in Sambrook et al., Molecular Cloning: A Laboratory Manual (New York: Cold Spring Harbor Laboratory Press, 1989). An alternative means to isolate the gene encoding SRT is to use PCR methodology [Sambrook et al., supra; Dieffenbach et al., PCR Primer: A Laboratory Manual (Cold Spring Harbor Laboratory Press, 1995)].

The Examples below describe techniques for screening a cDNA library. The oligonucleotide sequences selected as probes should be of sufficient length and sufficiently unambiguous that false positives are minimized. The oligonucleotide is preferably labeled such that it can be detected upon hybridization to DNA in the library being screened. Methods of labeling are well known in the art, and include the use of radiolabels like ^{32}P-labeled ATP, biotinylation or enzyme labeling. Hybridization conditions, including moderate stringency and high stringency, are provided in Sambrook et al., supra.

Sequences identified in such library screening methods can be compared and aligned to other known sequences deposited and available in public databases such as GenBank or other private sequence databases. Sequence identity (at either the amino acid or nucleotide level) within defined regions of the molecule or across the full-length sequence can be determined using methods known in the art and as described herein.

Nucleic acid having protein coding sequence may be obtained by screening selected cDNA or genomic libraries using the deduced amino acid sequence disclosed herein for the first time, and, if necessary, using conventional primer extension procedures as described in

Sambrook et al., supra, to detect precursors and processing intermediates of mRNA that may not have been reverse-transcribed into cDNA.

2. Selection and Transformation of Host Cells

Host cells are transfected or transformed with expression or cloning vectors described herein for SRT production and cultured in conventional nutrient media modified as appropriate for inducing promoters, selecting transformants, or amplifying the genes encoding the desired sequences. The culture conditions, such as media, temperature, pH and the like, can be selected by the skilled artisan without undue experimentation. In general, principles, protocols, and practical techniques for maximizing the productivity of cell cultures can be found in Mammalian Cell Biotechnology: a Practical Approach, M. Butler, ed. (IRL Press, 1991) and Sambrook et al., supra.

Methods of eukaryotic cell transfection and prokaryotic cell transformation are known to the ordinarily skilled artisan, for example, $CaCl_2$, $CaPO_4$, liposome-mediated and electroporation. Depending on the host cell used, transformation is performed using standard techniques appropriate to such cells. The calcium treatment employing calcium chloride, as described in Sambrook et al., supra, or electroporation is generally used for prokaryotes. Infection with *Agrobacterium tumefaciens* is used for transformation of certain plant cells, as described by Shaw et al., Gene, 23:315 (1983) and WO 89/05859 published 29 June 1989. For mammalian cells without such cell walls, the calcium phosphate precipitation method of Graham and van der Eb, Virology, 52:456-457 (1978) can be employed. General aspects of mammalian cell host system transfections have been described in U.S. Patent No. 4,399,216. Transformations into yeast are typically carried out according to the method of Van Solingen et al., J. Bact., 130:946 (1977) and Hsiao et al., Proc. Natl. Acad. Sci. (USA), 76:3829 (1979). However, other methods for introducing DNA into cells, such as by nuclear microinjection, electroporation, bacterial protoplast fusion with intact cells, or polycations, e.g., polybrene, polyornithine, may also be used. For various techniques for transforming mammalian cells, see Keown et al., Methods in Enzymology, 185:527-537 (1990) and Mansour et al., Nature, 336:348-352 (1988).

Suitable host cells for cloning or expressing the DNA in the vectors herein include prokaryote, yeast, or higher eukaryote cells. Suitable prokaryotes include but are not limited to eubacteria, such as Gram-negative or Gram-positive organisms, for example, Enterobacteriaceae

such as *E. coli*. Various *E. coli* strains are publicly available, such as *E. coli* K12 strain MM294 (ATCC 31,446); *E. coli* X1776 (ATCC 31,537); *E. coli* strain W3110 (ATCC 27,325) and K5 772 (ATCC 53,635). Other suitable prokaryotic host cells include Enterobacteriaceae such as *Escherichia*, e.g., *E. coli*, *Enterobacter*, *Erwinia*, *Klebsiella*, *Proteus*, *Salmonella*, e.g., *Salmonella typhimurium*, *Serratia*, e.g., *Serratia marcescans*, and *Shigella*, as well as *Bacilli* such as *B. subtilis* and *B. licheniformis* (e.g., *B. licheniformis* 41P disclosed in DD 266,710 published 12 April 1989), *Pseudomonas* such as *P. aeruginosa*, and *Streptomyces*. These examples are illustrative rather than limiting. Strain W3110 is one particularly preferred host or parent host because it is a common host strain for recombinant DNA product fermentations. Preferably, the host cell secretes minimal amounts of proteolytic enzymes. For example, strain W3110 may be modified to effect a genetic mutation in the genes encoding proteins endogenous to the host, with examples of such hosts including *E. coli* W3110 strain 1A2, which has the complete genotype *tonA* ; *E. coli* W3110 strain 9E4, which has the complete genotype *tonA ptr3*; *E. coli* W3110 strain 27C7 (ATCC 55,244), which has the complete genotype *tonA ptr3 phoA E15 (argF-lac)169 degP ompT kan*^{r}; *E. coli* W3110 strain 37D6, which has the complete genotype *tonA ptr3 phoA E15 (argF-lac)169 degP ompT rbs7 ilvG kan*^{r}; *E. coli* W3110 strain 40B4, which is strain 37D6 with a non-kanamycin resistant *degP* deletion mutation; and an *E. coli* strain having mutant periplasmic protease disclosed in U.S. Patent No. 4,946,783 issued 7 August 1990. Alternatively, *in vitro* methods of cloning, e.g., PCR or other nucleic acid polymerase reactions, are suitable.

In addition to prokaryotes, eukaryotic microbes such as filamentous fungi or yeast are suitable cloning or expression hosts for SRT-encoding vectors. *Saccharomyces cerevisiae* is a commonly used lower eukaryotic host microorganism. Others include *Schizosaccharomyces pombe* (Beach and Nurse, Nature, 290: 140 [1981]; EP 139,383 published 2 May 1985); *Kluyveromyces* hosts (U.S. Patent No. 4,943,529; Fleer et al., Bio/Technology, 9:968-975 (1991)) such as, e.g., *K. lactis* (MW98-8C, CBS683, CBS4574; Louvencourt et al., J. Bacteriol., 737 [1983]), *K. fragilis* (ATCC 12,424), *K. bulgaricus* (ATCC 16,045), *K. wickeramii* (ATCC 24,178), *K. waltii* (ATCC 56,500), *K. drosophilarum* (ATCC 36,906; Van den Berg et al., Bio/Technology, 8:135 (1990)), *K. thermotolerans*, and *K. marxianus; yarrowia* (EP 402,226); *Pichia pastoris* (EP 183,070; Sreekrishna et al., J. Basic Microbiol., 28:265-278 [1988]); *Candida; Trichoderma reesia* (EP 244,234); *Neurospora crassa* (Case et al., Proc. Natl. Acad. Sci. USA, 76:5259-5263 [1979]); *Schwanniomyces* such as *Schwanniomyces occidentalis* (EP

394,538 published 31 October 1990); and filamentous fungi such as, e.g., *Neurospora, Penicillium, Tolypocladium* (WO 91/00357 published 10 January 1991), and *Aspergillus* hosts such as *A. nidulans* (Ballance et al., Biochem. Biophys. Res. Commun., 112:284-289 [1983]; Tilburn et al., Gene, 26:205-221 [1983]; Yelton et al., Proc. Natl. Acad. Sci. USA, 81: 1470-1474 [1984]) and *A. niger* (Kelly and Hynes, EMBO J., 4:475-479 [1985]). Methylotropic yeasts are suitable herein and include, but are not limited to, yeast capable of growth on methanol selected from the genera consisting of *Hansenula, Candida, Kloeckera, Pichia, Saccharomyces, Torulopsis,* and *Rhodotorula*. A list of specific species that are exemplary of this class of yeasts may be found in C. Anthony, The Biochemistry of Methylotrophs, 269 (1982).

Suitable host cells for the expression of glycosylated SRT are derived from multicellular organisms. Examples of invertebrate cells include insect cells such as Drosophila S2 and Spodoptera Sf9, as well as plant cells. Examples of useful mammalian host cell lines include Chinese hamster ovary (CHO) and COS cells. More specific examples include monkey kidney CV1 line transformed by SV40 (COS-7, ATCC CRL 1651); human embryonic kidney line (293 or 293 cells subcloned for growth in suspension culture, Graham et al., J. Gen Virol., 36:59 (1977)); Chinese hamster ovary cells/-DHFR (CHO, Urlaub and Chasin, Proc. Natl. Acad. Sci. USA, 77:4216 (1980)); mouse sertoli cells (TM4, Mather, Biol. Reprod., 23:243-251 (1980)); human lung cells (W138, ATCC CCL 75); human liver cells (Hep G2, HB 8065); and mouse mammary tumor (MMT 060562, ATCC CCL51). The selection of the appropriate host cell is deemed to be within the skill in the art.

3. Selection and Use of a Replicable Vector

The nucleic acid (e.g., cDNA or genomic DNA) encoding SRT may be inserted into a replicable vector for cloning (amplification of the DNA) or for expression. Various vectors are publicly available. The vector may, for example, be in the form of a plasmid, cosmid, viral particle, or phage. The appropriate nucleic acid sequence may be inserted into the vector by a variety of procedures. In general, DNA is inserted into an appropriate restriction endonuclease site(s) using techniques known in the art. Vector components generally include, but are not limited to, one or more of a signal sequence, an origin of replication, one or more marker genes, an enhancer element, a promoter, and a transcription termination sequence. Construction of suitable vectors containing one or more of these components employs standard ligation techniques which are known to the skilled artisan.

The SRT may be produced recombinantly not only directly, but also as a fusion polypeptide with a heterologous polypeptide, which may be a signal sequence or other polypeptide having a specific cleavage site at the N-terminus of the mature protein or polypeptide. In general, the signal sequence may be a component of the vector, or it may be a part of the SRT-encoding DNA that is inserted into the vector. The signal sequence may be a prokaryotic signal sequence selected, for example, from the group of the alkaline phosphatase, penicillinase, lpp, or heat-stable enterotoxin II leaders. For yeast secretion the signal sequence may be, e.g., the yeast invertase leader, alpha factor leader (including *Saccharomyces* and *Kluyveromyces* α-factor leaders, the latter described in U.S. Patent No. 5,010,182), or acid phosphatase leader, the *C. albicans* glucoamylase leader (EP 362,179 published 4 April 1990), or the signal described in WO 90/13646 published 15 November 1990. In mammalian cell expression, mammalian signal sequences may be used to direct secretion of the protein, such as signal sequences from secreted polypeptides of the same or related species, as well as viral secretory leaders.

Both expression and cloning vectors contain a nucleic acid sequence that enables the vector to replicate in one or more selected host cells. Such sequences are well known for a variety of bacteria, yeast, and viruses. The origin of replication from the plasmid pBR322 is suitable for most Gram-negative bacteria, the 2μ plasmid origin is suitable for yeast, and various viral origins (SV40, polyoma, adenovirus, VSV or BPV) are useful for cloning vectors in mammalian cells.

Expression and cloning vectors will typically contain a selection gene, also termed a selectable marker. Typical selection genes encode proteins that (a) confer resistance to antibiotics or other toxins, e.g., ampicillin, neomycin, methotrexate, or tetracycline, (b) complement auxotrophic deficiencies, or (c) supply critical nutrients not available from complex media, e.g., the gene encoding D-alanine racemase for *Bacilli.*

An example of suitable selectable markers for mammalian cells are those that enable the identification of cells competent to take up the SRT-encoding nucleic acid, such as DHFR or thymidine kinase. An appropriate host cell when wild-type DHFR is employed is the CHO cell line deficient in DHFR activity, prepared and propagated as described by Urlaub et al., Proc. Natl. Acad. Sci. USA, 77:4216 (1980). A suitable selection gene for use in yeast is the *trp*1 gene present in the yeast plasmid YRp7 [Stinchcomb et al., Nature, 282:39 (1979); Kingsman et al., Gene, 7:141 (1979); Tschemper et al., Gene, 10:157 (1980)]. The *trp*1 gene provides a selection

marker for a mutant strain of yeast lacking the ability to grow in tryptophan, for example, ATCC No. 44076 or PEP4-1 [Jones, Genetics, 85:12 (1977)].

Expression and cloning vectors usually contain a promoter operably linked to the SRT-encoding nucleic acid sequence to direct mRNA synthesis. Promoters recognized by a variety of potential host cells are well known. Promoters suitable for use with prokaryotic hosts include the β-lactamase and lactose promoter systems [Chang et al., Nature, 275:615 (1978); Goeddel et al., Nature, 281:544 (1979)], alkaline phosphatase, a tryptophan (trp) promoter system [Goeddel, Nucleic Acids Res., 8:4057 (1980); EP 36,776], and hybrid promoters such as the tac promoter [deBoer et al., Proc. Natl. Acad. Sci. USA, 80:21-25 (1983)]. Promoters for use in bacterial systems also will contain a Shine-Dalgarno (S.D.) sequence operably linked to the DNA encoding SRT.

Examples of suitable promoting sequences for use with yeast hosts include the promoters for 3-phosphoglycerate kinase [Hitzeman et al., J. Biol. Chem., 255:2073 (1980)] or other glycolytic enzymes [Hess et al., J. Adv. Enzyme Reg., 7:149 (1968); Holland, Biochemistry, 17:4900 (1978)], such as enolase, glyceraldehyde-3-phosphate dehydrogenase, hexokinase, pyruvate decarboxylase, phosphofructokinase, glucose-6-phosphate isomerase, 3-phosphoglycerate mutase, pyruvate kinase, triosephosphate isomerase, phosphoglucose isomerase, and glucokinase.

Other yeast promoters, which are inducible promoters having the additional advantage of transcription controlled by growth conditions, are the promoter regions for alcohol dehydrogenase 2, isocytochrome C, acid phosphatase, degradative enzymes associated with nitrogen metabolism, metallothionein, glyceraldehyde-3-phosphate dehydrogenase, and enzymes responsible for maltose and galactose utilization. Suitable vectors and promoters for use in yeast expression are further described in EP 73,657.

SRT transcription from vectors in mammalian host cells is controlled, for example, by promoters obtained from the genomes of viruses such as polyoma virus, fowlpox virus (UK 2,211,504 published 5 July 1989), adenovirus (such as Adenovirus 2), bovine papilloma virus, avian sarcoma virus, cytomegalovirus, a retrovirus, hepatitis-B virus and Simian Virus 40 (SV40), from heterologous mammalian promoters, e.g., the actin promoter or an immunoglobulin promoter, and from heat-shock promoters, provided such promoters are compatible with the host cell systems.

Transcription of a DNA encoding the SRT by higher eukaryotes may be increased by